



**Strathmore**  
UNIVERSITY

Strathmore University  
**SU+ @ Strathmore**  
University Library

---

**Electronic Theses and Dissertations**

---

2017

# A Mobile based rent collection system

James Ngugi Wangari  
*Faculty of Information Technology (FIT)*  
*Strathmore University*

Follow this and additional works at <https://su-plus.strathmore.edu/handle/11071/5715>

## Recommended Citation

Wangari, J. N. (2017). *A Mobile based rent collection system* (Thesis). Strathmore University. Retrieved from <http://su-plus.strathmore.edu/handle/11071/5715>

This Thesis - Open Access is brought to you for free and open access by DSpace @ Strathmore University. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of DSpace @ Strathmore University. For more information, please contact [librarian@strathmore.edu](mailto:librarian@strathmore.edu)

# **A Mobile Based Rent Collection System**

Wangari James Ngugi

Submitted in partial fulfilment of the requirements for the Degree of Masters of Science in Mobile Telecommunication and Innovation (MSc. MTI) at Strathmore University

Faculty of Information Technology  
Strathmore University  
Nairobi, Kenya

June, 2017

This dissertation is available for Library use on the understanding that it is copyright material and that no quotation from the dissertation may be published without proper acknowledgement.

## **Declaration**

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the dissertation contains no material previously published or written by another person except where due reference is made in the dissertation itself.

© No part of this thesis may be reproduced without the permission of the author and Strathmore University.

James Ngugi Wangari 092499

Signature .....

Date.....

### **Approval**

The dissertation of James Ngugi Wangari was reviewed and approved by the following:

Dr. Humphrey Njogu,

Faculty of Information Technology

Strathmore University

Dr. Joseph Orero,

Dean, Faculty of Information Technology,

Strathmore University.

Prof. Ruth Kiraka,

Dean, School of Graduate Studies,

Strathmore University.

## **Acknowledgments**

I acknowledge, with gratitude, my debt of thanks to Safaricom Academy and Strathmore University for an opportunity to undertake my master's degree in their program. Special acknowledgements to Wainaina and Joan for infecting me with the art of questioning everything and Dr. Njogu for whose guidance I am truly grateful.

## **Dedication**

To the memory of my grandmother Jacinta Wairimu Ngugi, a gentle soul with an extra-ordinary mind and a giving heart. To the family and friends who paved the path I tread on; whose giant shoulders I stand upon. To my mother, for everything. To a certain *her* for always being there. And to Jeff whose memory lives on.

## **Abstract**

More than a third of Kenyans live in urban centers, majority of whom live in individual rental units. The proliferation of mobile phones and mobile money services in these urban centers surpasses 80% and 91.6% respectively. However, majority of these tenants still pay rent using traditional methods either by cash or through bank deposits. Generally, bank deposits take time before reflecting on the property owners' end while cash payments require property managers to meet with their tenants each month for the collection of rent. Majority of property owners are not able to track their tenants' payments when the payment method is bank deposits or cash payment.

The main focus of this study is to use mobile technologies to design and develop a mobile based rent collection system. The solution offers real time updates to managers about their tenants rent payments. The study used the Agile Software methodology to design and develop a mobile application that can be used by property owners and managers to collect and track their tenants' rent payments. Tenants on the other hand can access their payment details by accessing a USSD application. The application affords tenants the ability to pay for their rent by mobile money, in whatever amounts and frequency they may choose. The proposed solution recorded high performance and the transactions are reflected almost instantly on the property owner's mobile application after payment. The study concludes by recommending the adoption of the proposed solution.

**Keywords:** Rent Collection, Mobile Money, Payments, USSD

## Table of Contents

Declaration.....	i
Acknowledgments.....	ii
Dedication.....	iii
Abstract.....	iv
List of Figures.....	ix
List of Tables.....	x
List of Abbreviations/Acronyms.....	xi
Chapter 1: Introduction.....	1
1.1 Background of the Study.....	1
1.2 Problem Statement.....	2
1.3 Research Objectives.....	2
1.4 Research Questions.....	2
1.5 Research Justification.....	2
1.6 Scope of the Study.....	3
Chapter 2: Literature Review.....	4
2.1 Introduction.....	4
2.2 The Housing Sector.....	4
2.3 Methods Used To Collect Revenues.....	6
2.4 Challenges of Rent Collection in Kenya.....	7
2.5 Technology and Revenue Collection.....	8
2.5.1 <i>Electronic Funds Transfer (EFT)</i> .....	8
2.5.2 <i>Real-Time Gross Settlement System (RTGS)</i> .....	8
2.5.3 <i>Mobile Money</i> .....	9
2.5.4 <i>Electronic Credit Cards</i> .....	9
2.5.5 <i>Electronic Cash</i> .....	10
2.6 Existing Rent Collection Solutions.....	10
2.6.1 <i>Cozy Services</i> .....	10
2.6.2 <i>PayYourRent</i> .....	12
2.6.3 <i>RentPayment</i> .....	13

2.6.4 <i>Lipa Kodi</i> .....	14
2.6.4 <i>MRent</i> .....	14
2.7 Conceptual Model .....	15
2.8 Summary .....	15
Chapter 3: Research Methodology .....	17
3.1 Introduction .....	17
3.2 Research Design .....	17
3.3 Software Development Methodology .....	17
3.3.1 <i>Requirements Discovery</i> .....	18
3.3.2 <i>System Design</i> .....	18
3.3.3 <i>System Development</i> .....	19
3.3.4 <i>System Testing and Validation</i> .....	19
3.4 Research Quality Aspects .....	20
3.4.1 <i>Reliability</i> .....	21
3.4.2 <i>Validity</i> .....	21
3.5 Summary .....	21
Chapter 4: System Design .....	22
4.1 Introduction .....	22
4.2 Functional Requirements .....	22
4.3 Non-Functional Requirements .....	22
4.4 System Architecture .....	23
4.4.1 <i>Data Input Capture</i> .....	23
4.4.2 <i>Processing Module</i> .....	24
4.4.3 <i>Reporting and Analytics Module</i> .....	24
4.5 General Algorithms .....	25
4.5 Database Design .....	25
4.6 Other Design Tools .....	28
4.6.1 <i>Data Flow Diagram</i> .....	29
4.6.2 <i>Class Diagram</i> .....	32
4.6.3 <i>Sequence Diagram</i> .....	33

4.6.4 Use Case Diagram.....	34
4.7 Network Design .....	34
4.8 Security Design.....	35
4.9 Application Wireframes.....	36
4.9.1 Application Backend.....	37
4.9.2 Mobile Application.....	39
4.9.3 Tenant USSD.....	40
4.10 Summary.....	41
Chapter 5: System Development and Testing.....	42
5.1 Introduction.....	42
5.2 Software Environment .....	42
5.3 Hardware Environment.....	42
5.4 Modules of the System.....	42
5.5.1 USSD Application .....	43
5.5.2 Backend Web Application.....	43
5.5.3 Mobile Application.....	45
5.5 Testing And Validation.....	46
5.5.1 Functional Testing.....	47
5.5.2 Usability Testing.....	48
5.5.3 Load Testing.....	49
5.5.4 Compatibility Testing.....	49
5.5.5 Integration Testing.....	50
5.5.6 Validation.....	50
5.6 Conclusion .....	53
Chapter 6: Discussion of Results .....	54
6.1 Mobile-Based Rent Collection System.....	54
6.2 Advantages of the Application.....	55
6.3 Limitations of the Proposed Application .....	55
Chapter 7: Conclusions, Recommendations & Future Work.....	56
7.1 Conclusion .....	56

7.2 Recommendation .....	56
7.3 Future Work .....	57
References.....	58
Appendix A: Application Evaluation Questionnaire .....	61
Appendix B: Extra Application Wireframes.....	63
Appendix C: Extra Application Screenshots .....	64
Appendix D: Turnitin Report.....	65

## List of Figures

Figure 2.1 Cozy Services Architecture .....	11
Figure 2.2 PayYourRent System Architecture.....	12
Figure 2.3 RentPayment System Architecture.....	13
Figure 2.4 Conceptual Model .....	15
Figure 3.1 Agile Software Methodology .....	17
Figure 4.1 System Architecture .....	23
Figure 4.2 System Entity Relationship Diagram .....	25
Figure 4.3 System Context Diagram.....	29
Figure 4.4 System Data Flow Diagram Level 1 .....	30
Figure 4.5 System Class Diagram.....	32
Figure 4.6 Transaction Processing Sequence Diagram.....	33
Figure 4.7 System Use Case diagram .....	34
Figure 4.8 Application Client-Server Architecture.....	35
Figure 4.9 Tenants Page Wireframe .....	37
Figure 4.10 Notices Page Wireframe.....	38
Figure 4.11 Payments Page Wireframe .....	38
Figure 4.12 Mobile Application Wireframes.....	39
Figure 4.13 USSD Application Wireframe.....	40
Figure 5.1 USSD Application Screenshot One.....	43
Figure 5.2 USSD Application Screenshot Two .....	43
Figure 5.3 Web Application Property Page Screenshot 1.....	44
Figure 5.4 Web Application Rooms Page Screenshot 2 .....	44
Figure 5.5 Web Application Payments Page Screenshot.....	45
Figure 5.6 Mobile Application Screenshot One .....	46
Figure 5.7 Mobile Application Screenshot Two.....	46
Figure 5.8 Load Testing Results .....	49
Figure 5.9 Application Validation Questionnaire Results (a).....	51
Figure 5.10 Application Validation Questionnaire Results (b) .....	52
Figure 5.11 Application Validation Questionnaire Results (c).....	53

## **List of Tables**

Table 4.1 Users Table .....	26
Table 4.2 Property Table.....	27
Table 4.3 Transactions Table .....	27
Table 4.4 SMS Table .....	28
Table 5.1 Functional Testing Results.....	47
Table 5.2 Usability Testing Results .....	48
Table 5.3 Mobile Application Compatibility Testing Report.....	50
Table 5.4 Browser Compatibility Testing Report.....	50

## **List of Abbreviations/Acronyms**

API- Application Programming Interface

B2B – Business to Business

B2C – Business to Customer

CBK – Central Bank of Kenya

DFD – Data Flow Diagram

ERD –Entity Relationship Diagram

GSM -Global System for Mobile Communications

KRA- Kenya Revenue Authority

KNBS- Kenya National Bureau of Statistics

OOA- Object Oriented Analysis

OOD -Object-Oriented Design

P2P- Person to Person

PHP- Hypertext Preprocessor

SMS- Short Messaging Service

SSD – System Sequence Diagram

USSD- Unstructured Supplementary Service Data

# Chapter 1: Introduction

## 1.1 Background of the Study

According to a 2016 Kenya Bureau of Statistics, more than a third of Kenyans live in urban centers, (KNBS, 2016): more than half (57.4%) of whom live in individual rental units. The rest live in houses by the government, local authority, parastatal or private company (KNBS, 2016). Only about one-third (34%) of the households live in their own houses – purchased, constructed or inherited. The city of Nairobi has the largest population closely followed by Mombasa, Kisumu and Eldoret. Rent, therefore, is a recurring bill for a majority of urban dwellers. In this dissertation, tenants are defined as households paying a prearranged rent for the exclusive occupation of all or part of a house (Arvanitis, 2015). Additionally, this dissertation does not distinguish between types of property owners. A property owner is taken to be any individual or institution that owns accommodation that they rent out (Arvanitis, 2015).

According to the same study, the proliferation of mobile phones and mobile money services in urban centers was also very high, estimated to be more than 80% and 91.6% (Kemibaro, 2016) respectively. The majority of these tenants only pay rent by cash or through bank deposits (Kinaku, 2011). However, bank deposits require a follow up on the tenant's part to ensure the property owner has received and updated their payment while cash payments require property managers to meet with their tenants each month for the collection of rent.

The use of these traditional methods of payment – cash and bank deposits, has numerous challenges to both the property manager and the tenant. Some property managers use other systems to manage and track their tenant's details and payment history (DeSmit, 2014). In such cases, when the tenant makes a payment using the above mentioned methods, the property manager has to update the tenant's details in their system. This usually leads to payment details reflecting late on their systems and more often input errors generated in the update process (Eberlin, 2016). These payment methods are therefore tedious and error prone. Property managers have limited ways of accessing, in real-time, their tenants rent payments details.

A study by Noppen (Noppen, 2013) a researcher from the Acumen foundation, proposed the use of new technologies to automate rent collection. However, most of these systems (as discussed in chapter 2) do not support mobile money services such as Mpesa, Airtel Money or EazzyPay as a

payment option (Alleman & Rappoport, 2010). These systems, nonetheless, have had various successes in other parts of the world (Cozy Services Ltd).

## **1.2 Problem Statement**

The traditional methods of rent payment - cash and bank deposits, have two main challenges. Generally bank deposits take time before reflecting on the property owners end after a tenant makes a payment and especially on weekends and holidays (DeSmit, 2014). Cash payments, on the other hand, usually obligate a property owner to meet their tenant to facilitate rent collection. Additionally, the property owner has to manually maintain a record of every tenant's payment. Majority of property owners, therefore, are generally not able to track their tenants' payments in real-time (Eberlin, 2016).

Based on aforementioned challenges, this study sought to design a mobile based rent collection system that allows tenants to pay rent by mobile money. The system offers property owners a mobile application that allows them to track their tenants' payments in real-time.

## **1.3 Research Objectives**

- i. To identify common rent payment collection methods.
- ii. To review existing mobile based rent payment collection systems.
- iii. To design, develop and test a mobile based rent payment collection system.

## **1.4 Research Questions**

- i. What are the common rent payment collection methods?
- ii. What are the existing mobile-based rent payment collection systems?
- iii. How can a mobile-based rent payment collection system be designed, developed and tested?

## **1.5 Research Justification**

This study intends to explore the various rent payment collection systems in use and propose how a mobile based rent payment collection system can be developed. In particular, this study explored the use of mobile money as a rent payment method. The primary product of this study is an application that runs on Android mobile phones. The application can be used by property

owners and property managers to track the payments of their tenants. This application supports mobile money payments as the primary method of payment. The application also allows property owners to keep track of their tenants rent payments in real time. Tenants are be able to pay rent using Mpesa. They are also able to access their payment history using a USSD application in order to keep track of their rent arrears.

In addition, the mobile application affords property managers the ability to send notices to their tenants and the ability for tenants to send complaints to the property managers through SMS. As a consequence, tenants have ease and efficiency while paying rent from their mobile phones. Tenants can additionally pay their rent in small amounts and at their comfort and time.

## **1.6 Scope of the Study**

This study only focuses on rent payment within Langata constituency of Nairobi County due to the high population rate and mobile phone penetration in this county. The study only adopted the Mpesa mobile money service as the rent payment method.

## **Chapter 2: Literature Review**

### **2.1 Introduction**

This chapter presents a detailed overview the current rent methods and systems being employed various areas of the world. 4 systems employed in other parts of the world for rent collection are discussed against the background of statistics on the rate of property ownership in Nairobi. The chapter relies heavily on a study conducted by the Central Bank of Kenya (CBK), the Kenya National Bureau of Statistics (KNBS) in the period between 2012 and 2016 to show how many urban dwellers live in rental units. Finally, a conceptual model of the proposed system is discussed.

### **2.2 The Housing Sector**

As of 2012, the Kenyan population growth was estimated at 4.2% per annum (CBK, 2016), majority of whom were living in urban areas. Based on this growth and the rate of urban migration, the yearly annual increase in demand for housing in Kenya is at 206,000 units annually 82,000 of which is in urban areas (Arvanitis, 2015). In 2011 the Ministry of Housing estimated that the formal supply of houses to the market reached 50,000 creating a 156,000 shortfall. On the other hand, over a span of 11 years from 2001 to 2012, the average prices for 1 to 3 bedroom houses rose by a factor of 2 from just below Ksh.5 million (USD 60,000 at 2001 rates) to Ksh.10 million (USD 117,000 at 2012 rates). Prices for units with 4 to 6 bedrooms rose from about Ksh.10 million to Ksh.31 million (USD 362,000 at 2012 rates) (Arvanitis, 2015).

Nationally, more than half (57.4%) of the urban households lived in rental units according to the Kenya National Bureau of Statistics (CBK, 2016). Only about one-third (34 percent) of the households lived in their own house whether purchased, constructed or inherited (KNBS, 2016). These findings were established after a survey of 230 urban centers. The study also found out that the proportion of people living in urban centers had increased to 15.1 percent in 1979 and to 31.3 percent in 2009.

Of all the urban centers under study, the Nairobi County had the largest population followed closely by Mombasa. In an analysis of the adult population (18 years and above) in the county by the Central Bank of Kenya (2016), more than half (53.7%) of the city dwellers were found to

have attained secondary education, 80% had access to a mobile phone and close to a fifth (18%) of the phones had Internet access (CBK, 2016). An overwhelming 91.6% of the participants had used a mobile money service such as Mpesa, Airtel Money or Yu-Mobile. The same survey established that about 60% of the city dwellers earned an average monthly income in the range of Ksh.30,000 or less, of which a maximum of about a third was usually allocated for rent payment.

According to a report by the UN-HABITAT (2003), rent is considered by many property owners to be an income that is more tangible than putting money into a bank or other kind of financial institution. According to the report, property owners rent for a variety of reasons. Either as a safety net against precarious employment, meeting household expenditure, housing improvements, a regular source of income when moving from waged employment to own forms of employment or as a form of pension after retirement and old age and as investment for the next generation. In other cases sometimes, room letting provides more than a basic subsistence. For some landlords rentals represent a way of financing the completion of the main house, for others they provide a means to pay taxes and fees to get property rights. Many landlords would argue that the real value of owning property and renting it out is the security it offers for the future. After all, the same report says, large numbers of landlords are old and are living either on their pension or entirely from the proceeds of the rents.

Kenyan pay rent by either cash, through bank deposits or through cash transfer by mobile money services such as Mpesa or Airtel Money (Kinaku, 2011). However, these methods are riddled by challenges such as late payment and difficulty doing follow-ups when the number of tenants is large (DeSmit, 2014). While many property owners use other systems to manage various property aspects, few systems exist that allow them to accept payments through mobile money. These methods are discussed in more detail in the next section.

In a study aimed at understanding the supply-demand mismatch found in the Kenyan housing sector, Noppen (2013) proposed three main avenues through which to address the problems of housing and rent payment. In particular, Noppen emphasized on the need to build more houses, the importance of financing the end-user to afford a house, or the invention of new technologies to facilitate easier payment of rent and mortgage installments.

## **2.3 Methods Used To Collect Revenues**

Currently, three main methods of collecting revenue/rent are in use; payment by cash, bank deposit or through mobile money cash transfers (Kinaku, 2011). These methods are discussed in detail in the following section.

### ***2.3.1 Cash***

This is done where the tenants handover their rent remittances as cash to the property owner or manager. The tenant is then issued with a payment receipt. As Xinhua (2014) states, the tenant should therefore be in a position to visit the property managers office to handover the money. A study by FSD (2012) found out that cash payments are especially popular in informal settlements where the rent ranges are between 3000 and 10000 shillings.

### ***2.3.2 Direct Bank Deposit***

Other tenants pay rent through bank deposits to a specific bank account as directed by their property manager. Once the tenant has deposited the rent, they take the bank deposit slip to their property manager for payment confirmation and clearance (Xinhua, 2014). In other instances, a tenant may conduct an inter-bank transfer from their account to the property owners account (FSD, 2012).

### ***2.3.3 Mobile Money Transfers***

Finally, some property owners allow their tenants to send remittances through mobile money services such as Mpesa, Airtel money and Equitel Eazzy pay. Xinhua (2014) notes that, since the property owners usually have their tenants phone numbers and personal details, they can be able to determine which tenant sent how much money and when. As the study from FSD noted, smaller and independent property management companies accept mobile money transactions under the person-to-person (P2P) modality. This is usually because most of them do not have Paybill numbers (for Mpesa) and business numbers (for Airtel Money) to receive mobile money payments. Instead, tenants typically send rent as a person-to-person (P2P) transaction to the property managers. However, the tenant is still expected to visit the property managers' office to collect their payment receipt.

### ***2.3.3 Automatic Bill Payment***

An Automatic Bill Payment refers to a money transfer scheduled on a predetermined date to pay a recurring bill. Automatic bill payments are routine payments made from a banking account to vendors. Automatic payments can be made from a checking account or credit card. They are usually set up with the company receiving the payment, though it's also possible to schedule automatic payments through a checking account's online bill pay service. Automatic bill payments occur over an electronic payment system, such as the Automated Clearing House (ACH). Regular payments can be automated quite easily from a checking account by making arrangements with the bank holding the checking account to make the exact payment each month. The account holder decides whether the automatic bill payment happens each month; he or she has the power to turn the automatic payment function off or to postpone the payment.

## **2.4 Challenges of Rent Collection in Kenya**

Cash payments and bank deposits have both pros and cons. DeSmit (2014) recommends against accepting rent by cash. DeSmit argues that cash is easy to lose, hard to trace, and there may be discrepancies on how much the tenant paid versus how much the property owner received. In particular, a major disadvantage of cash payment is that each month, the property manager will need to meet with the tenant for the collection of rent.

On the other hand, direct bank deposits require a tenant to walk into an ATM or banking hall, withdraw money from their account, and then queue at the specified bank to deposit or goes to the property manager's office to pay (Mwakilishi, 2016). Mwakilishi continues to state how this process is not only time-consuming and inconveniencing, but also insecure.

Mobile money offers an easy way for tenants to transfer money from their phone to their property manager. However, when used as a cash transfer method in the context of rent payment this method suffers from the same disadvantages as direct bank deposits (Dolan, 2009). When the number of tenants is large, greater than 20 according to Eberlin (2016), it becomes time-consuming and frustrating to try and coordinate payment times with all the tenants.

In particular, for property owners who use an automated way to track their tenants' payments, bank deposits and cash and mobile money transfers reconciliation of payment details with their existing records is a major challenge (FSD, 2012). As Eberlin (2016) and Mwakilishi (2016) explain, these manual update processes result to tenant payments reflecting late on their systems

and subtle input errors maybe introduced in the process. Finally, all these methods require a trip to the property managers' office to collect a payment receipt. This can be a time-consuming endeavor for tenants who do not live near their property managers.

## **2.5 Technology and Revenue Collection**

This section describes how technology has so far been leveraged to enable revenue collection in Kenya. Electronic payments in particular are discussed based on the technologies they use. In this section, electronic payments are to be considered to be transactions taking place through electronic equipment using various types of trading cards, computer technology or communication technology as a means of issuing payment instructions directly or indirectly to achieve payment or the transfer of funds (He, Shi, & He, 2010).

### ***2.5.1 Electronic Funds Transfer (EFT)***

Popularly known as direct deposit, the EFT is a system that allows a user to make payments by transferring money from one bank account directly to another without any paper money changing hands, such as depositing salaries into employee's bank accounts. According to Nakhumwa (2013), EFT has expanded to refer to any transfer of money initiated through an electronic terminal, including credit/ debit smart cards, Automated Teller Machine (ATM), Electronic Funds Transfer at Point of Sale (EFTPOS), Electronic Data Interchange (EDI) and Internet banking. A user makes a payment by authorizing an inter-bank cash transfer from their account to another account. These payments are processed through the Nairobi Automated Clearing House (ACH) managed by the Kenyan Bankers' Association (KBA). The payments are batched, and funds take two days to clear between banks. The EFT levies a small charge per transaction. The charges depend on transaction size, but are typically be in the KShs.50-300 range (FSD, 2012).

### ***2.5.2 Real-Time Gross Settlement System (RTGS)***

The RTGS is a system operated by the Central Bank of Kenya that allows large inter-bank payments to be made. The RTGS system facilitates the payment of amounts, which are larger than what the EFT can handle. RTGS supports continuous concurrent processing and final settlement of funds transfer instructions from one bank to another, in the accounts of participants in the Central Bank of Kenya as long as they have sufficient covering balance or credit Large

payments of over Kshs 1 million must be routed through the Kenya Electronic Payment and Settlement System (KEPSS), though smaller payments are also possible. KEPSS is operated by the Central Bank of Kenya (CBK), and all banks are connected to it. Though the settlement is immediate within banks, it may take 2-6 hours for banks to reflect smaller payments on their customers' accounts (FSD, 2012).

### ***2.5.3 Mobile Money***

Janine (2015) defines mobile money as a service in which the mobile phone is used to access financial services. Mobile Money is the stored money value held on a cellular mobile device that could be used in paying for goods or services, sent to another subscriber or converted back to cash (Alleman & Rappoport, 2010). In mobile money, a mobile phone number is used as an account. Funds are stored in a secure electronic account linked to the mobile phone number (William & Suri, 2011). Paying with mobile money is just like sending a text message. To pay a bill or send money to another person, the user selects the relevant service from their phone's mobile money menu, selects or inputs the recipients' phone number and amount and submits the request. What differentiates mobile money services from other payment methods is this simple, low-tech mechanism for providing money transfers.

The Kenya mobile money market is dominated by Mpesa, a service operated by mobile operator Safaricom. A survey by Mbiti & Weil (2011) estimated this service to have over 15 million users (82% of the Safaricom customer base) who were able to cash in/out at some 30,000 locations across the country. Competing mobile money services have emerged in Kenya, some managed by smaller mobile operators (such as Airtel Money, Orange Money and YuCash), and some operated by independent players (such as MobiKash or Tangaza). Banks are also launching mobile banking and payments capabilities (such as Equity BankEazzy247 service and KCB Mobile Banking), supported by the spread of thousands of banking agents at shops (CBK, 2016). All these services allow users to send money to other users, to pay for goods at specific retail outlets and to bank accounts.

### ***2.5.4 Electronic Credit Cards***

Credit and smart cards are a commonly used method of electronic payment and are widely accepted by consumers and merchants throughout the world, especially in retail markets

(Gharegozi, Faraji, & Heydari, 2011). A credit card is a small plastic card issued to users as a method of payment for online or off-line purchases. These cards are issued by financial institutions, and authorized cardholders can take it as payment tool to pay for consumer billing to the traders. For credit cards, a consumer credit is generally used in advance to set a spending limit, and consumers can pay the balance on the card and spend a certain amount of overdraft. Online credit card payment systems are very suitable for Business to Consumer (B2C) model and small size Business to Business (B2B) model (Yifei, 2008). Credit card payment system adopts registered consumption patterns, which makes the overdraft possible and the security of the system get strengthened. However it loses the feature of anonymity, so it cannot properly protect the privacy; therefore it is not suitable as a mode of person to person payment.

### ***2.5.5 Electronic Cash***

E-cash, also known as digital cash, is a digital form of currency in circulation through the network (Yifei, 2008). It converts cash's numerical value into a series of encrypted numeric sequence of numbers to represent the various amounts of currency in reality through these sequences. Consumers open bank accounts in banks where have e-cash business, deposit money in the account, and then they can shop in the stores where accept electronic cash. E-cash applies the secure and reliable digital signature with good security; payment process needs not always rely on the financial network (i.e. off-line payments), which is low cost and suitable for small payments; it can be used anonymously and the process of use cannot be tracked. However, the anonymity of electronic cash and non-traceability of electronic cash makes the holders not be able to report missing, once their information is lost.

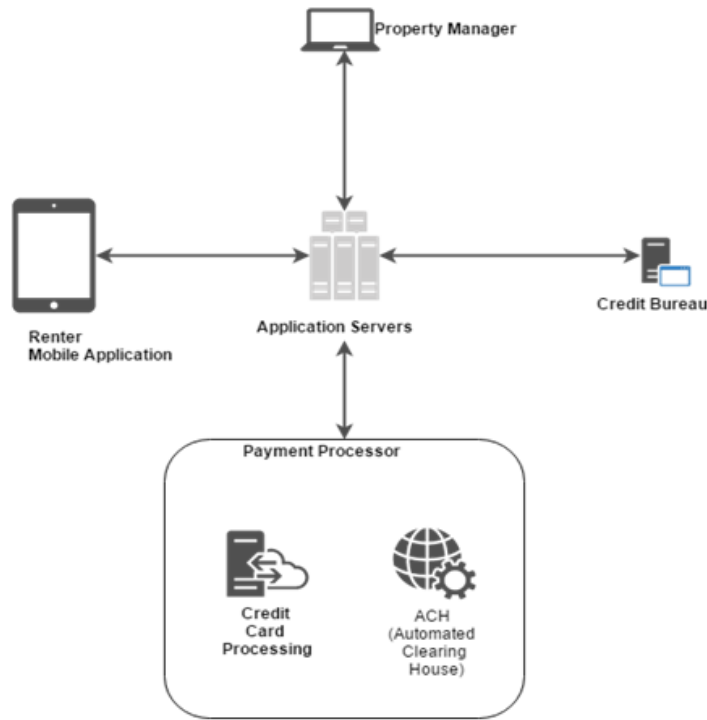
## **2.6 Existing Rent Collection Solutions**

Paying monthly bills online is increasingly gaining foothold in Kenyan households (Irungu, 2014). Rent, being a major monthly household bill, has not been left behind. New payment platforms have now made it possible for tenants to remit rent directly without going the traditional way (Kinaku, 2011). Four such platforms are discussed in this section.

### ***2.6.1 Cozy Services***

Cozy is a property management for landlords, property managers, and renters offered by Cozy Services Ltd (Cozy Services Ltd). The company is based in Vancouver Portland but their services

are available to companies in the United States. Cozy offers modules for both tenants and property owners. The service offers property managers services such as property listing on partner advertising platforms, online rental applications from tenants, tenant screening, credit reports and background checks.

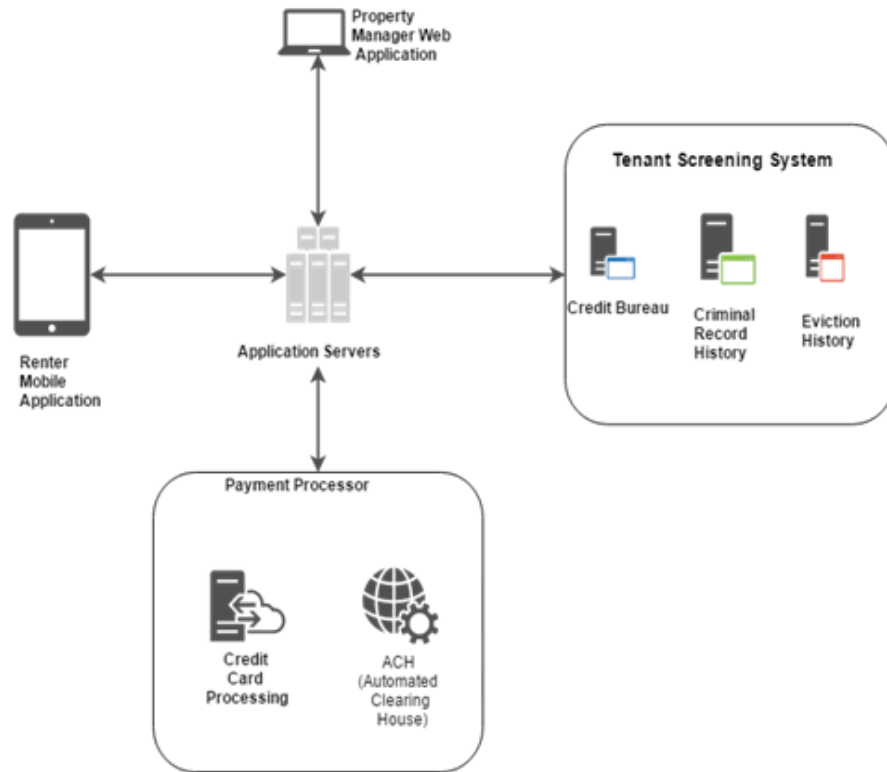


**Figure 2.1 Cozy Services Architecture**

As shown in Figure 2.1, the platform has a payment processing module responsible for credit card processing and bank deposits processing. The system also interfaces with a credit bureau to ascertain the credit worthiness of each renter. The platform allows tenants to apply for tenancy, and to optionally provide a credit report in the process. Additionally, property can conduct background checks and request for screening reports from each of their tenants. The service is available as a web application. Tenants can pay for rent using their credit card or their bank account. Payment processing typically takes 2-4 business days (Paycorp) to be fully complete. However, this service requires each tenant to download the mobile application that runs on the iOS platform, in order to make a payment.

### 2.6.2 PayYourRent

PayYourRent is an online rent payments processing based in Beverly Hills, California (PayYourRent). The service offers property management services to property managers and landlords and online rent payment services to tenants and residents.



**Figure 2.2 PayYourRent System Architecture**

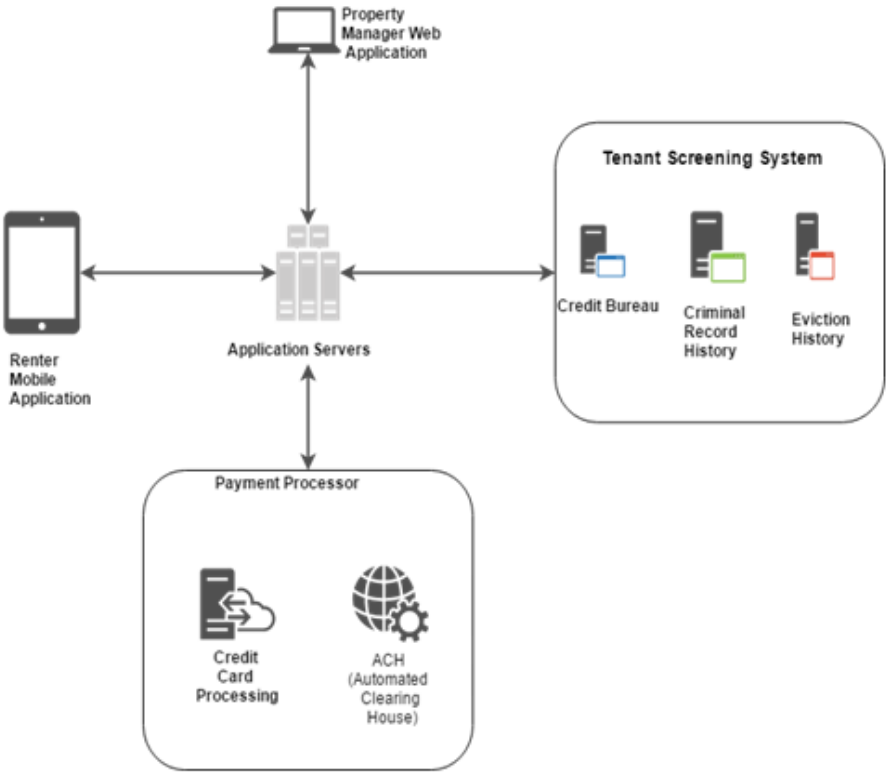
As shown in Figure 2.2, the service is available as a web application for property managers and an Android application and an iOS application for tenants, landlords and property owners. The platform interfaces with external databases such as credit bureau, criminal records and evictions databases to offer property managers the ability to screen tenants before they offer them tenancy. Tenants can pay their rent using a bank account or credit card. They also receive services such as the ability to request property maintenance, to schedule recurring payments and to setup payment reminders.

The company has partnered with utility companies to allow residents to connect all utilities and home services with the most affordable pricing. Tenants can compare local and national brands

then setup water, power, gas, cable, phone, and the Internet services according to their budget. Tenants can pay for rent using their credit cards or bank accounts. However, the service does not currently support payments by mobile money services.

**2.6.3 RentPayment**

RentPayment is a California based company that offers online rent payment (RentPayment). The company allows property owners to add their properties and their tenants. Tenants can then pay for rent using their credit card, debit card or using a custom payment gateway called eCheck.



**Figure 2.3 RentPayment System Architecture**

As shown in Figure 2.3 the RentPayment platform is offered as a mobile application that runs on the Android and iOS platforms for both tenants and property while property owners have access to a web application. Tenants can review their payment history on the application or submit payments. The platform interfaces with external databases such as the credit bureau such that when tenants pay for rent via the platform, their credit worthiness is increased. Consequently, they get lower interest rates for their future loans. The platform also interfaces with MoneyGram

Cash transfer service to allow tenants to pay rent by cash in addition to credit cards, debit cards and bank transfers.

However, due to the low uptake of credit cards and smart cards in Kenya this service can only be used by very tenants in the Kenyan context (Nakhumwa, 2013) .

#### ***2.6.4 Lipa Kodi***

Lipa Kodi is a service offered by Safaricom (Matinde, 2013) that enables house rent payers to submit their monthly dues through mobile money payment Mpesa. The service is charged at Kshs 5 from rent amount of Kshs 5,000 to Kshs 220 for upwards of Kshs 70,000 rent. This service targets the small income earners who usually pay through cash or have to do cash deposits. According to Matinde (2013), the service was aimed at selling the idea to home agents and owners. He continues to argue that this service was introduced to complement other Mpesa related services including, Lipa Na Mpesa, in order to recruit businesses, small and large to use Safaricom's services.

The service offers property owners and managers a web application where they can manage and access their tenants' payments. Renters can only pay rent. Property owners are required to own an Mpesa Paybill account through which they can receive the rent payments (FSD, 2012). FSD continues to argue that, the web tool that Safaricom offers its corporate customers to view their transactional histories does not permit easy visualization or searching of transactions. Only transactions in the last six months can be accessed. This has limited the uptake of these services among corporate property managers and owners.

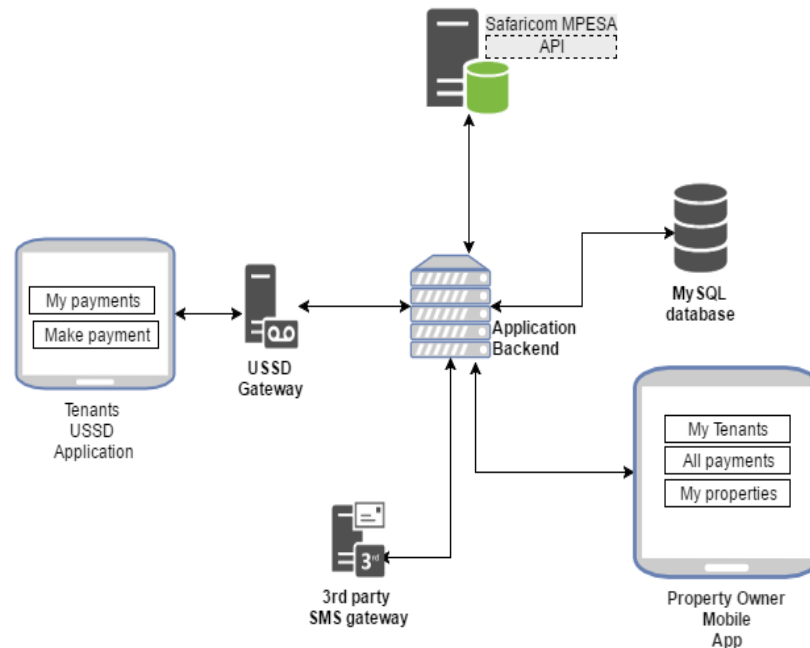
#### ***2.6.4 MRent***

MRent offers a basic web-based property management services to property managers, enabling them to manage several properties at no charge. Through mobile payment platforms such as M-Pesa, Airtel Money, Orange Money and yuCash, a tenant can send money to a bill number allocated to the landlord or property manager from a mobile phone. The funds are automatically deposited into the owner's bank account within 72 hours. The transactions are done at no extra cost besides the normal network charges for withdrawals. With MRent, both the tenant and property manager can access up-to-date payment records and transactions online besides instant confirmation through SMS whenever a transaction is made. The property manager can also view

different forms of payments made to the account and even manual adjustments made to the system. The MRent system also allows real estate agents to add properties, manage tenant units and collect rent on units set up for online rent payment.

## 2.7 Conceptual Model

The proposed study will be based on a conceptual model shown in figure 2.4. The application has two interfaces, a USSD application and a mobile application.



**Figure 2.4 Conceptual Model**

The USSD application is used by tenants while the mobile application is used by property owners. These are backed by a backend application that connects to a relational database, an Mpesa payment gateway and an SMS gateway. The application backend runs a web application which is also accessible to property owners. A MySQL database is used to store all system information. The application backend connects to the Safaricom MPESA API to actuate the payment of rent by MPESA.

## 2.8 Summary

This chapter introduced a statistical view of the housing sector. By highlighting the rate of urbanization against the backdrop of population growth in Kenya, the housing demand was also

discussed. The chapter then proceeded to show how a significant number of the urban population live in individual rental units, and therefore consider rent a major monthly bill. A discussion of the various ways in which technology is used to collect revenues and the current methods of rent collection then follows. In the same context, four existing systems that are used to manage rent collection were discussed. These systems are evaluated on the basis of supported payment techniques, supported platforms among other features. In particular, the offering of mobile money as a payment method is either not fully supported or not at all supported in the evaluated systems. By highlighting these systems and challenges, the first two research questions were fully addressed. This chapter finally illustrated and discussed a conceptual model of a mobile-based rent collection system that uses mobile money. The proceeding chapter therefore will seek to answer the last two research questions by discussing how a mobile application that supports mobile money as a payment method can be developed and validated.

## Chapter 3: Research Methodology

### 3.1 Introduction

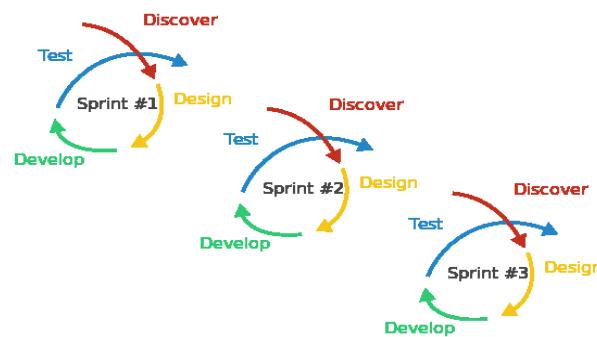
This chapter focuses on the Software Development Methodology used to design the mobile based rent collection system. It then proceeds to discuss the research methodology to be used in carrying out the research in line with the research questions. Finally, the research quality aspects are discussed.

### 3.2 Research Design

The study will employ an exploratory research design to explore fundamental insights into the rent collection process. This will also help establish a deeper look into the technologies of mobile money services, as well as the security of their related transactions. The identified features will also form a basis for crafting the system tests for the final developed system.

### 3.3 Software Development Methodology

This used the Agile Software development methodology. The proposed system underwent 4 main phases as shown in Figure 3.1 and discussed in the next section.



**Figure 3.1 Agile Software Methodology adapted from Munassar & Govardhan (2010)**

Agile Software Development is built on the premise that software development is inherently uncertain (Munassar & Govardhan, 2010) and that changes should be expected. The researcher assumed that the project mobile application would be fully designed, developed and tested within the stipulated academic time. Additionally, in the course of development, the researcher identified new functional requirements that could be added to the proposed application. An agile

approach was adopted where the proposed software was built in iterations, each iteration adding a specific feature to the mobile application. Agile accommodated such requirement changes by allowing easy correction in the course development.

Agile Methodology suited this research appropriately because, as outlined by (Munassar & Govardhan, 2010); changes could be made to the applications and not have such a long impact, the methodology encourages participation from clients and hence user acceptance is almost guaranteed and unlike the Waterfall and other old Methodologies where the requirements are to be stated at the beginning of development, then the development divided into phases and assigned to various developers who would discuss after completion of those phases. Agile helps to remove this by making communication easy among the teams and changes that should be made are easy.

### ***3.3.1 Requirements Discovery***

During this phase, the researcher sought to know various functional requirements and the non-functional requirements of the system. A detailed and thorough review of existing documents on the structure of mobile money services and rent collection to understand the existing rent collection systems and to help formulate a solution that could work with existing rent collection systems. This was then documented to facilitate the functionality requirements for the development of the proposed mobile application.

### ***3.3.2 System Design***

A Data Flow Diagram (DFD) was used to show a graphical contextual design of the system. This clarified any external systems interacting with the system. Object oriented analysis was used to identify and model all objects interacting with the system. These objects relationships were also identified. Use Case diagrams and use case descriptions were then used to model each object's interaction with the system. Using the Use Cases enabled the researcher to separate the system into actors and use cases. Each use case was represented accompanied by a description that described the action the user is carrying out on the system.

Finally, an Entity Relationship Diagram (ERD) was used to graphically conceptualize the relationships between the identified objects. The ERD was also used to model the application

database which contained each entity and its attributes in a relational database. In addition, high level design mock-ups of the system user interfaces were constructed using the Balsamic mockup tool. This tool was used to prototype the interfaces of the proposed mobile and web applications.

### ***3.3.3 System Development***

This research aimed at coming up with a mobile application and a web application. Both the mobile and web application were connected to a central database. The mobile application was developed for the Android operation system using the Java programming language. JavaScript Object Notation (JSON) was used as the data transfer format between the web application and the mobile phone.

The USSD application for the tenants was developed using PHP and communicating with a MySQL backend. The USSD service allowed tenants to view and verify their rent balance and payments. The application communicated with a backend API implemented on the Laravel PHP framework. The backend application exposed a payment gateway, a web service which interacted with the Safaricom Mpesa API via Simple Object Access Protocol (SOAP), to authorize and authenticate rent payments. Finally, the backend API interacted with a third party SMS gateway to actuate SMS delivery.

### ***3.3.4 System Testing and Validation***

Once development of each module, testing was done on it. This was done to ensure the application meets the functional needs. To test the prototype functionalities and its user acceptance, various types of test were subjected on it as described below:

#### **i. Compatibility Testing**

Both the mobile and web applications were tested with different Android operating system (versions) and different web browsers respectively. This ensured the applications run on different Android and Web browsers for compatibility. The system was also expected to interface with other existing systems that property manager's use. This testing ensured that there is seamless interoperability between the systems. For instance, the transaction details were required to be exportable to the QuickBooks software commonly used by property managers.

## **ii. Load Testing**

This was done to check the amount of time the application takes to process a request. Load testing was conducted by generating and sending a high number of requests to the web application using a tool such as LoadView. The response rate was then be used to determine how many requests the application could handle during peak usage.

## **iii. Integration Testing**

The system was developed using a module-based approach. This test was therefore conducted on the system to know whether when the modules operate as desired when joined together. This was done by combining all modules of the prototype and testing if they worked together well without any issue. A unit-testing tool known as PHPUnit (Bergmann, 2017) was used to conduct unit testing to ensure the web application modules are working properly. This test helped to ensure integrations with the Mpesa API were successful. Additionally, as the web application was modified in iterations, a tool known as Selenium tools was used to conduct regression testing to ensure the web application remained functional.

## **iv. Validation**

This test was done to determine whether the application introduced efficiency into the rent collection process. The application was tested to check whether it offered a minimalistic interface that enables them accomplish tasks with minimal effort. Stratified random sampling was used to come up with 7 property owners who were requested to use the developed prototype. They were then given a questionnaire in order to get feedback as well as having talks with them to understand their experience. The feedback was then be analyzed to offer insight into the effectiveness of the application.

### **3.4 Research Quality Aspects**

Research quality, by definition, refers to the extent to which a research was carried out properly (NCDDR, 2005). In this study, research quality is assessed based on the validity and reliability of the research.

### ***3.4.1 Reliability***

Reliability is defined as the extent to which the outcomes of a research can be replicated, given the same conditions and approach (Kinaku, 2011). The principle idea of reliability is that the parameters of the population under study are stable or constant and should produce the same (or nearly the same) results if the same individuals and conditions are used. Once the proposed application is finalized, the researcher will conduct full testing to ascertain the validity of the proposed application in order to confirm whether the application introduces efficiency into the rent collection process. Such validity checks will ensure that the study can be replicated to produce a similar application in any future studies.

### ***3.4.2 Validity***

Kinaku (2011) defines validity as the extent to which a study's results truthfully and accurately reflect the intended objectives. In order to ascertain internal validity, the researcher intends to conduct a preliminary review of only the very recent literature and systems on the topic at hand. This preliminary review will ensure that the system to be implemented is developed in a way that can be integrated with the current systems that property managers use.

## **3.5 Summary**

This chapter has described the various phases the proposed mobile application underwent. A description of the tools employed in each of the phases is discussed in detail showing how each tool will be used. A justification of the agile methodology is offered as well as the research design to be employed. This chapter and the one proceeding therefore seek to answer the final two research questions by developing and testing a mobile application to automate the collection of rent and validating its effectiveness.

## **Chapter 4: System Design**

### **4.1 Introduction**

This chapter presents the analysis, design, implementation and testing procedures adopted in the development of the prototype proposed in the study. The architecture of the developed prototype, its algorithms, and process and database design are discussed in detail. Application wireframes of the prototype interfaces are then presented.

### **4.2 Functional Requirements**

The system is expected to offer the following key functionalities:

- i. Rent payment: The rent payment functionality allows tenants to pay for rent using their mobile phone. This functionality should be accessible by dialing a USSD shortcode.
- ii. Property details management: The system should allow property owners to add details about their properties, including rooms and their room numbers and details about the various tenants who occupy these rooms.
- iii. Complaint reporting: The system should allow tenants to send complaints by SMS to the property manager. These complaints should be accessible by the property owner through the mobile and web application.
- iv. Notice sending: The system should allow property managers to send notices to their tenants through SMS.

### **4.3 Non-Functional Requirements**

Non-functional requirements are the requirements that do not affect the core business of the application, the application can still work with or without them, but which are part of the system.

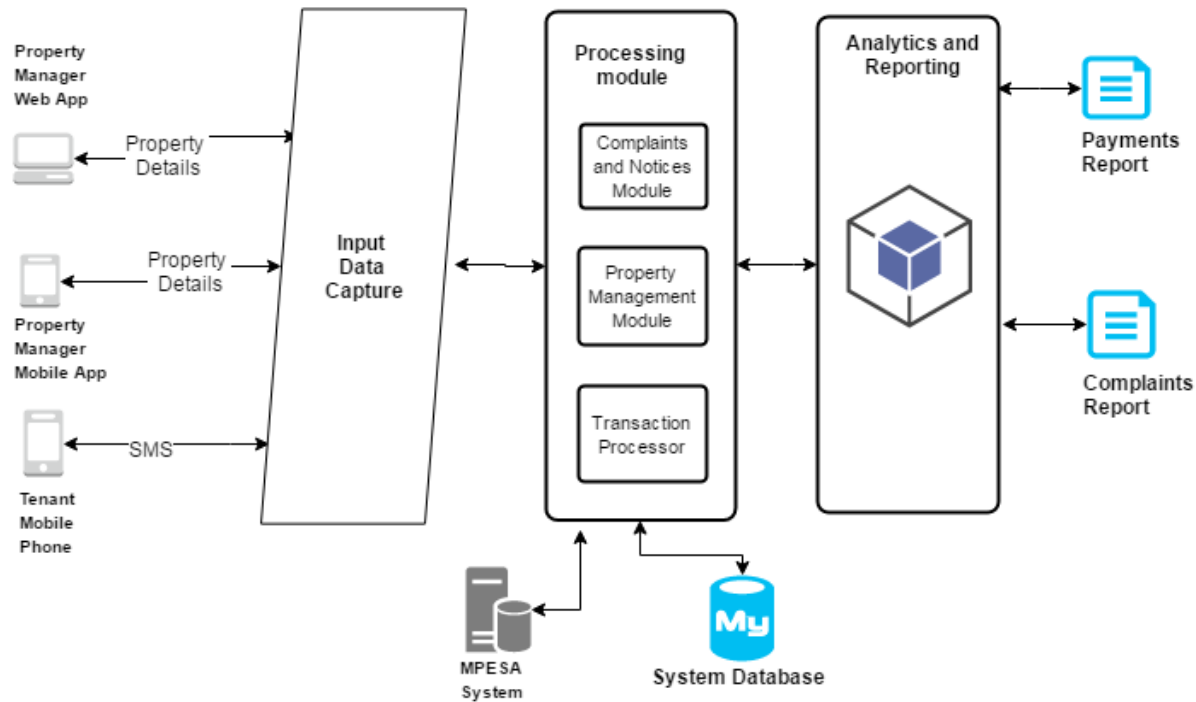
The non-functional requirements of the proposed system were:

- i. Usability: The system should have an easy to use interface.
- ii. Reliability and availability: The system should be reliable and always available to perform tasks.
- iii. Performance: The system should have an acceptable response time while performing functions.

- iv. Scalability: The system’s performance should not be adversely affected as the number of users’ increases.

## 4.4 System Architecture

The major components of the prototype are as shown in Figure 4.1. These components are discussed in detail in the following sections.



**Figure 4.1 System Architecture**

### 4.4.1 Data Input Capture

This module is responsible for collecting user input. The user input modules are dependent on the users’ application environment. For the tenants the user input is through the means of a USSD menu which is accessible by dialing a USSD short code. Property owners provide inputs through an interface running on a mobile application or a web application. Regardless of the input source, the user input module offers a uniform interface where the internal system modules can access user supplied data. Inputs from external services such as the SMS gateway and the payment gateway are normalized for easier access with the internal system components.

#### ***4.4.2 Processing Module***

The processing module is responsible for conducting processing on the input data. It has sub modules which perform various functions as outlined below:

The transaction processor is responsible for initiating, requesting and negotiating payments with the Safaricom MPESA gateway. Once a tenant indicates intent to make a payment through the USSD application, the transaction processor opens a connection to the payment gateway, which in turn prompts the user for their MPESA secret key through a USSD push. This module expects a valid Safaricom number, registered as an MPESA user and an amount that's within the transaction limit allowed by MPESA.

The property management module allows property owners to store and update details about the various properties, rooms and tenants from whom they wish to collect payments. This module offers an interface where users can feed these details into a form. It's only accessible by property owners on the web and mobile application. Property details include the property name and location, any rooms in the property including their room numbers while tenant details include the tenant name and phone number and the room where each tenant resides. The tenant phone number must be a Safaricom phone number. The module allows users to generate reports about all properties, rooms and tenants in either Portable Document Format (PDF) or spreadsheet formats.

The complaints module allows tenants to send any complaints or problems they encounter to their property managers through the application. Tenants can send problems by sending messages to an SMS shortcode. When an SMS is received from a tenant, the module automatically creates a complaint from it and informs the responsible property manager. The notices module, on the other hand, allows property managers to send alerts in form of SMS to tenants about any pertinent issues. These may include issues that affect individual houses or general information. Notices can be queued to be sent at later dates.

#### ***4.4.3 Reporting and Analytics Module***

The reports module aggregates data from the property management module, complaints module, notices module and payments module to display general reports about user payments per property. The reports are displayed in PDF and spreadsheet formats. The reports can be downloaded for offline reading.

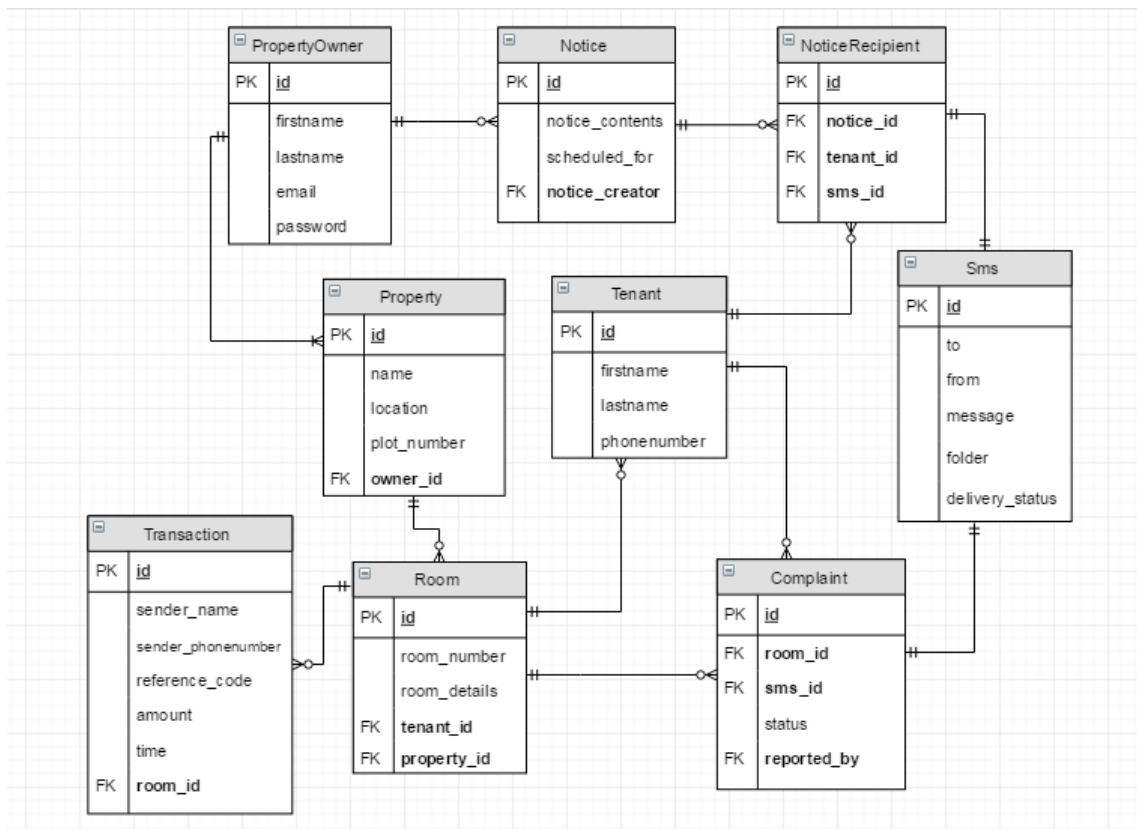
## 4.5 General Algorithms

When a tenant initiates a process to make a payment, the payment module sends a request to reserve a payment from the payment provider at Safaricom. The gateway then verifies whether the provided credentials are okay. The module then sends another request to confirm the transaction and move money from the tenant’s account to the systems account.

All messages received by the system are treated as complaints, for each message, the system checks whether message sender is a tenant in any room within the system before saving it. If the sender of the message is not a tenant, the message is ignored.

## 4.5 Database Design

The database system for the proposed system is represented using an Entity Relationship Diagram (ERD) as shown in Figure 4.2.



**Figure 4.2 System Entity Relationship Diagram**

Since the database was large, essential information was split into multiple tables. The data was normalized to avoid duplication. A user can either have the role of a property manager or a

tenant but not both. A property manager can add many properties which in turn can individually have multiple rooms. A room is expected to have only one tenant. Each payment, identified by transaction, pays for only one room. A tenant can send one or more complaints by SMS. Each complaint is tied to exactly one room and SMS. A property manager can send notices to all or selected tenants. Thus each notice can be sent to one or more recipients. Notices can be queued to be sent at a later date. Additionally, property owners must provide a unique email address and a password at least 6 characters long when registering. The database schema that details each of the entities attributes and primary keys is shown in the figures in the next section.

**Table 4.1 Users Table**

Column	Type	Null	Default	Comments
id	int(10)	No		
firstname	varchar(255)	No		
email	varchar(255)	No		
password	varchar(255)	No		
remember_token	varchar(100)	Yes	NULL	
created_at	timestamp	Yes	NULL	
updated_at	timestamp	Yes	NULL	
lastname	varchar(255)	No		
verified	tinyint(1)	No	0	
verification_token	varchar(255)	Yes	NULL	
phonenummer	varchar(50)	Yes	NULL	

Table 4.1 illustrates the users table. The id is the primary key; this is the unique identifier which identifies each user in the system. The password field contains an encrypted value of their password. The remember token field contains a value that tracks their login session while the verification token contains a token that is sent to the user so that they can activate their account. The verified field tracks whether or not they have activated their account.

The complaints table has a primary key id which uniquely identifies each complaint in the system. The complaint details field contains the complaint text received while the message\_id and room\_id identify the SMS message and room associated with this complaint respectively. The reported\_by field identifies the tenant who made the complaint. The status field contains information about whether the complaint has been opened by the property manager or not.

On the other hand, the notice contents field in the notices table contains the actual notice text that will be sent to the notice recipients. The scheduled for field contains the date when the notice should be sent. The notice creator contains the identifier of the property owner who posted the notice.

**Table 4.2 Property Table**

Column	Type	Null	Default	Comments
id	int(10)	No		
name	varchar(255)	No		
location	varchar(255)	Yes	NULL	
plot_number	varchar(255)	Yes	NULL	
manager_id	int(11)	No		

Table 4.4 shows the properties table. The id, the primary key, uniquely identifies each property in the system. The name, location and plot number identify the property details while the manager\_id field identifies the user who added the property to the system.

**Table 4.3 Transactions Table**

Column	Type	Null	Default	Comments
id	int(10)	No		
sender_ac_number	varchar(255)	Yes	NULL	
sender_ac_name	varchar(255)	Yes	NULL	
status	varchar(255)	No		
time	varchar(255)	No		
amount	double(8,2)	No		
transaction_reference	varchar(255)	No		
room_id	int(11)	Yes	NULL	

Table 4.5 illustrates the transactions table. This table contains details about each payment made through the system. The id, the primary key, uniquely identifies each property in the system. The sender\_ac\_number and sender\_ac\_name contain the payers MPESA phone number and full name respectively. The status field shows the status of the payment whether complete or pending. The amount details the amount paid. The transaction reference field contains the MPESA code of the payment. The room\_id contains the identifier of the room associated with this payment.

**Table 4.4 SMS Table**

<b>Column</b>	<b>Type</b>	<b>Null</b>	<b>Default</b>	<b>Comments</b>
id	int(10)	No		
to	varchar(255)	No		
from	varchar(255)	No		
message	varchar(255)	No		
sms_provider	varchar(255)	No		
time	datetime	No		
delivery_status	int(11)	Yes	<i>NULL</i>	
folder	varchar(255)	Yes	<i>NULL</i>	

The SMS table contains all messages sent in the system. Each message has a unique id attribute which identifies it within the system. The to field contains the phone number of the message recipient while the from field contains the message senders' phone number. The message field contains the actual message text. The sms\_provider field contains the name of the sms provider that transmitted the message. The time contains the date and time when the message was sent or received. The folder attribute identifies the message as incoming or outgoing.

#### **4.6 Other Design Tools**

This section discusses the various tools employed to design the modules of the proposed system. In particular, Data Flow Diagrams (DFDs) level 0 and level 1, Entity Relationship Diagrams including the data dictionary for each identified table, a class diagram based on the entities identified and a sequence diagram are discussed. Finally, a use case diagram that depicts the use cases of the system by various actors is shown.

#### 4.6.1 Data Flow Diagram

The flow of data in the proposed system is as shown in the Data Flow Diagram (DFD) shown in Figure 4.3.

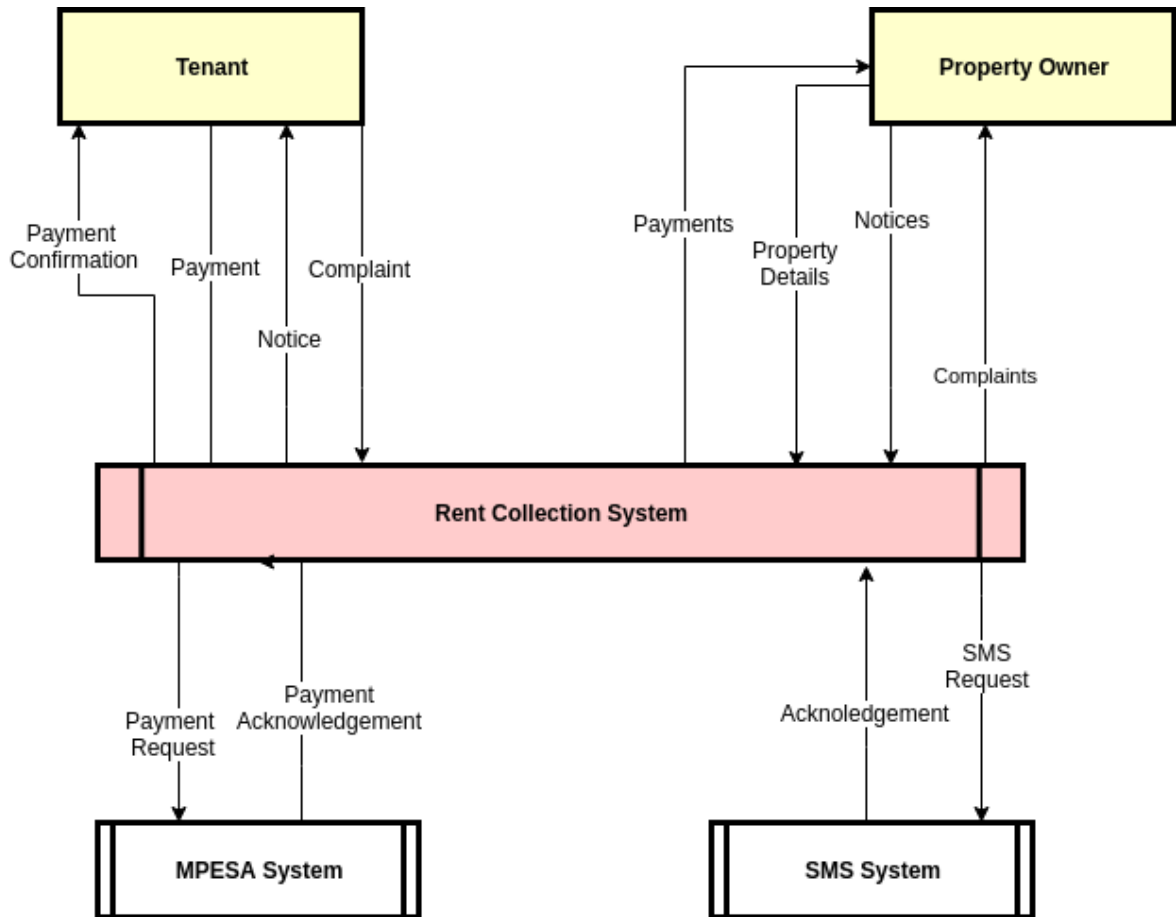
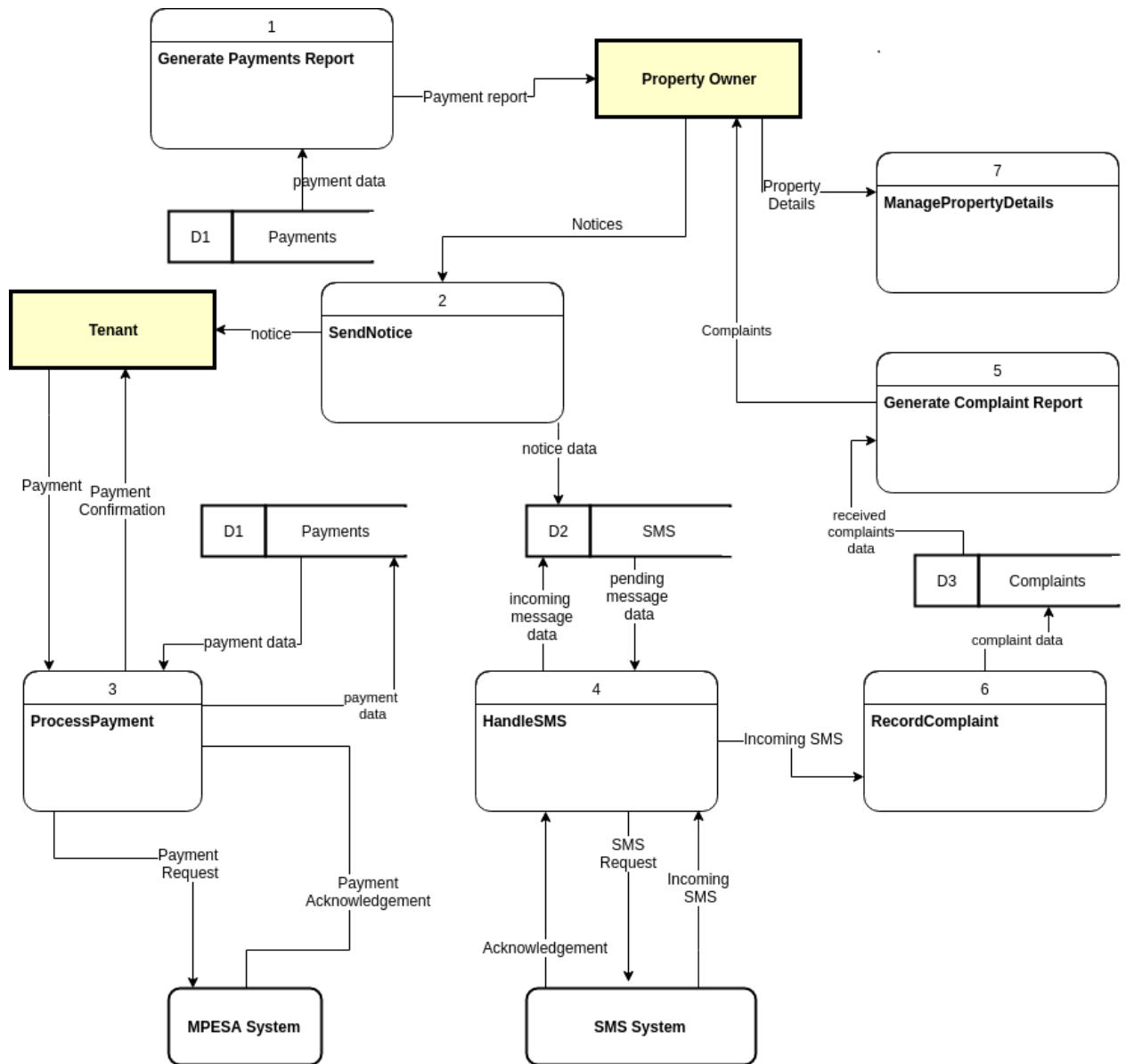


Figure 4.3 System Context Diagram



**Figure 4.4 System Data Flow Diagram Level 1**

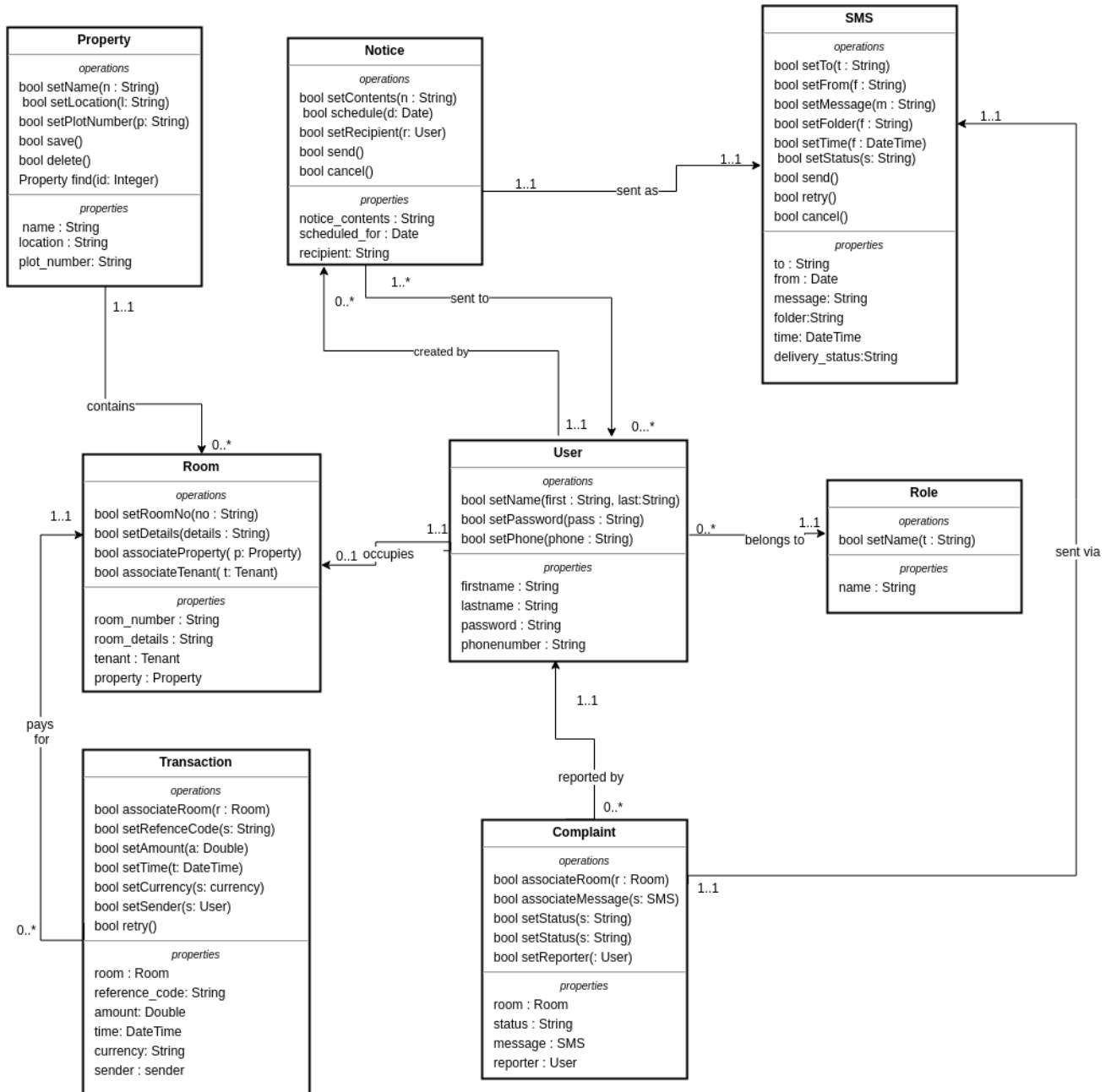
The system interacts with two external systems, a payment gateway offered by Safaricom MPESA and an SMS gateway that actuates USSD functionality and SMS sending. The two key users of the system are property owners and tenants.

The context diagram shows the context of the system as it interacts with various external systems. The DFD level 1 decomposes these interactions into several sub processes as shown in Figure 4.4 shows the various sub processes. The Process Payment sub process is responsible for negotiating and handling received payments. It interacts with the MPESA system and stores all

received payments in the Payments data store. The Generate Payments Report module is responsible for generating reports about all received payments. It uses the Payments data store. The Handle SMS sub process interfaces with the SMS gateway to send notices as SMS and to receive complains as SMS. The Manage Property Details allows property owners to add, edit and update property, room and tenant details.

The Record Complaint sub process saves any SMS received as complaints. It uses the complaints data store. The Send Notice sub process sends any scheduled notices by SMS. It uses the SMS data store. Finally, the Generate Complaint Report sub process generates detailed reports about any received complaints using data from the complaints data store.

## 4.6.2 Class Diagram

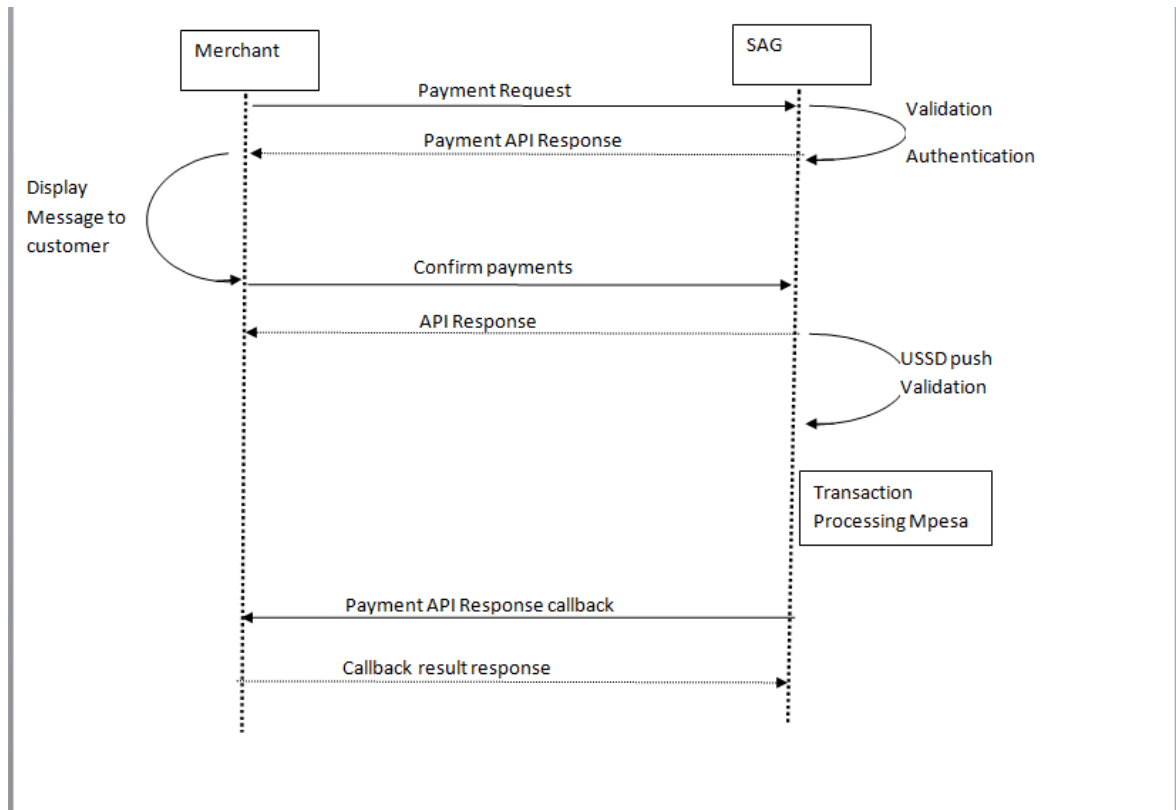


**Figure 4.5 System Class Diagram**

Based on the Entity Relationship Diagram, the class diagram of the proposed system is as shown in Figure 4.5. It shows the types being modeled within the system. The classifiers shown in this diagram are the objects. Each object is identified including its attributes (properties) and the methods it exposes for the manipulation of its attributes.

### 4.6.3 Sequence Diagram

The sequence diagram in Figure 4.6 shows the flow of events as a tenant wishes to make a payment. The merchant represents the payment processing module of the proposed application while SAG represents the MPESA Payment gateway. When a tenant initiates a request to make a payment, the merchant creates and sends payment request to the payment gateway requesting the initiating of a new payment.



**Figure 4.6 Transaction Processing Sequence Diagram**

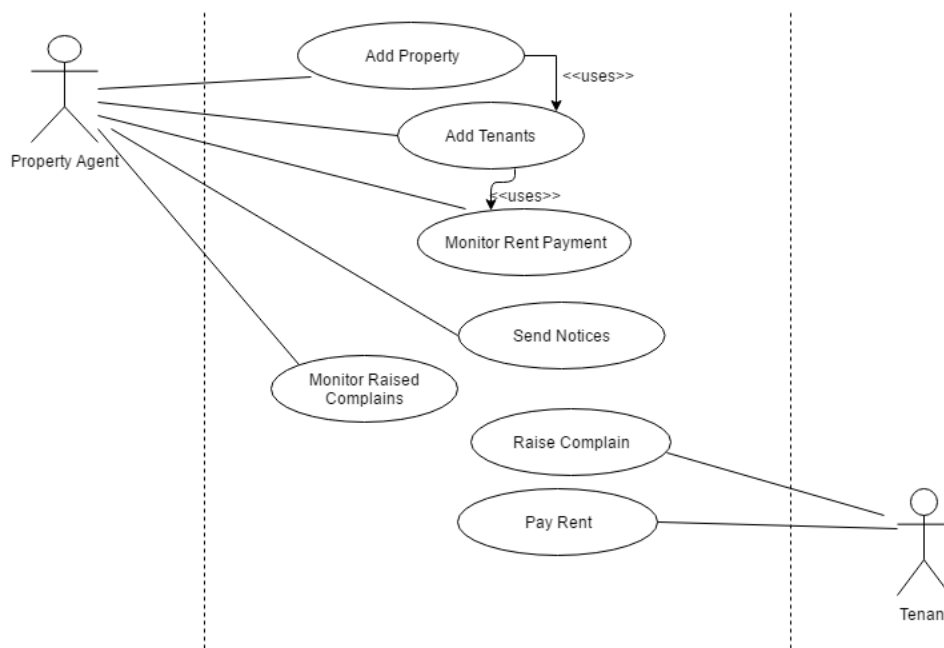
The gateway validates the request, ensures that the tenant has enough money in their MPESA account to make the payment, validates that the tenant is a registered MPESA user and authenticates that the merchant has provided valid credentials to make the request. The gateway then responds with message that indicates whether a payment has been successfully reserved and whether the merchant should continue with the payment.

If a payment was successfully reserved, the merchant sends a confirmation request in order to allow the payment gateway to proceed with the transaction. The gateway then sends a USSD prompt to the tenant's phone. The tenant is requested to provide their MPESA secret pin. Once

this is complete, the payment gateway moves money from the tenants account to the merchants account and sends a callback to the merchant with the complete transaction details. The merchant then settles the rent payment for the room in which the tenant currently lives in.

#### 4.6.4 Use Case Diagram

Figure 4.7 shows the system use case diagram for all the actors of the system. The system has two actors as stipulated in the previous sections: the property agent and tenant. The tenant can raise complaints, receive notices and pay rent while the property manager can add properties, rooms, tenants and monitor their payments. Additionally, they can monitor raised complaints and send notices to tenants.

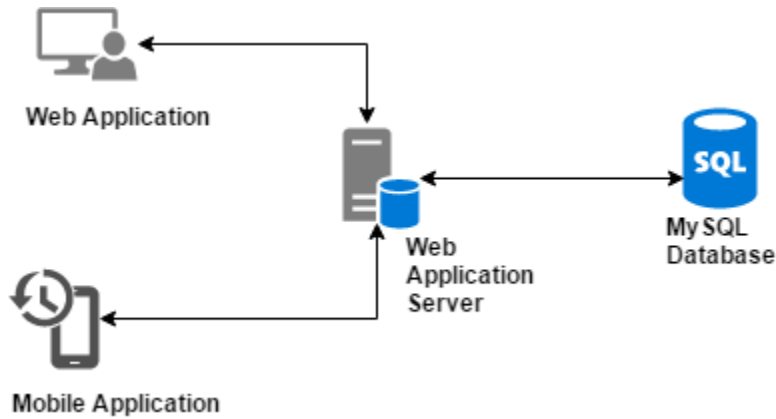


**Figure 4.7 System Use Case diagram**

#### 4.7 Network Design

The proposed system uses a client-server based network model for communication between the web application server and the mobile application and web application backend. Additionally, any changes made in the web application are broadcast to the mobile application using a publish-subscribe event model. Figure 4.8 shows the client-server architecture model adopted by the proposed application. The mobile application subscribes to changes from the web application. Whenever any entity or object changes state in the web application, the information about these

changes is broadcast to the mobile application. The mobile application then polls the web application to get a full list of what has changed.



**Figure 4.8 Application Client-Server Architecture**

## **4.8 Security Design**

The proposed application approaches security from two dimensions: security by policy and security by design. In security by policy, the application only requests for the absolutely essential pieces of information. Users are only required to input the information that is necessary. Additionally, all passwords are required to be at least 6 characters long. All users have at least one role either a tenant or property owner. All actions on the application are authenticated to ensure that each user has sufficient privileges to perform these tasks. Under security by design, various security risks are mitigated as discussed in the following section:

### **i. SQL Injection**

The web application uses PHP Data Object (PDO) (Zend Corp, 2017) parameter binding to avoid SQL injection. Parameter binding ensures that malicious users can't pass in query data which could modify the query's intent.

### **ii. Cross-Site Request Forgery**

CSRF (cross-site request forgery) tokens are used to ensure that third-parties cannot initiate malicious requests masquerading as legitimate users. This is done by generating a token that must be passed along with the form contents. This token are then be compared with a value

additionally saved to the user session. If it matches, the request is deemed valid, otherwise it is deemed invalid.

### **iii. Cross-Site Scripting**

All input is automatically escaped to remove HTML entities to avoid cross-site scripting. This ensures that any malicious code in the input does not cause any harm.

### **iv. API authentication**

Any communication between the mobile application and the web application is secured using OAuth2 (Bihis, 2015) authentication protocol. OAuth2 provides a nice way to authenticate mobile users via what is called token authentication. The OAuth2 token authentication works as follows: A user opens up a mobile application and is prompted for their username or email and password. The application sends a request from the mobile app to the web application service with the user's username or email and password data included, the user credentials are validated and an access token for the user that expires after a certain amount of time is generated. This access token is stored on the mobile device, treating it like an API key which lets the mobile application access the API service. Once the access token expires and no longer valid, the user re-prompted for their username or email and password.

### **v. Communication protocols**

The web application is delivered over Hypertext Transfer Protocol (HTTP); all cookies saved on the browser are encrypted to ensure they are secure.

### **vi. Encryption protocols**

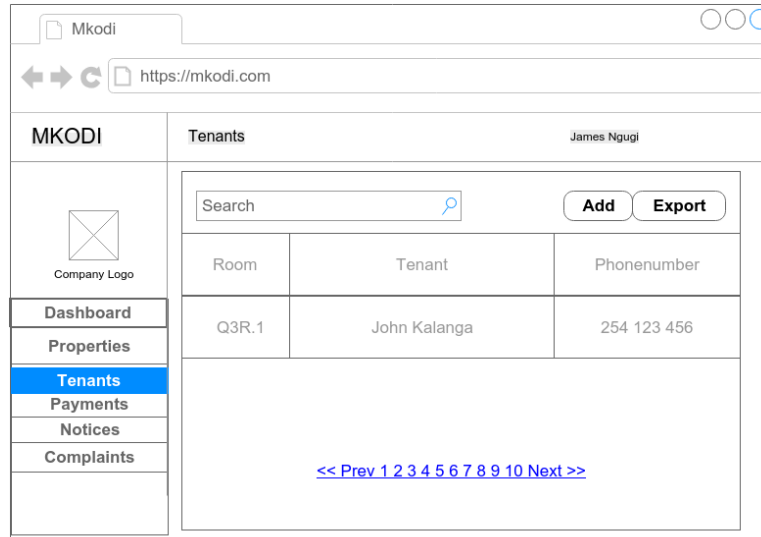
Passwords are encrypted using the bcrypt password hashing function, based on the blowfish cipher available on Linux systems (Provos & Mazières, 1999). Bcrypt incorporates a salt to protect against rainbow table attacks. It is also an adaptive function. Over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power. This ensures the passwords are secure.

## **4.9 Application Wireframes**

This section shows the user interface mockups for the proposed application. Since the application has three main interfaces namely: the USSD interface, the mobile application interface and the web application interface, this section is also subdivided into sections that relate to these modules.

### 4.9.1 Application Backend

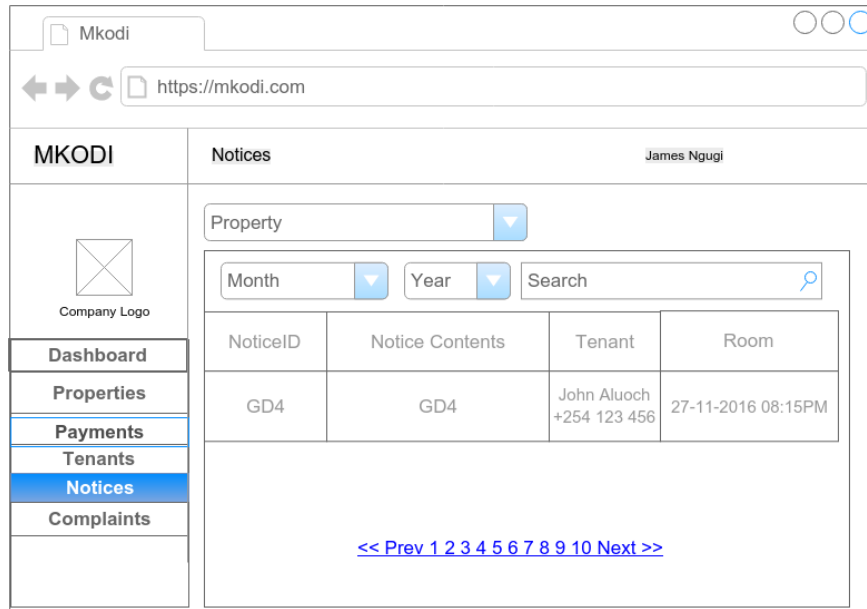
The application dashboard, shown in Appendix B, is the default screen property managers see after logging in to the web application. The properties page, allows users to add the various properties they manage to their ccount.They are prompted to provide the property name and plot number to proceed.



**Figure 4.9 Tenants Page Wireframe**

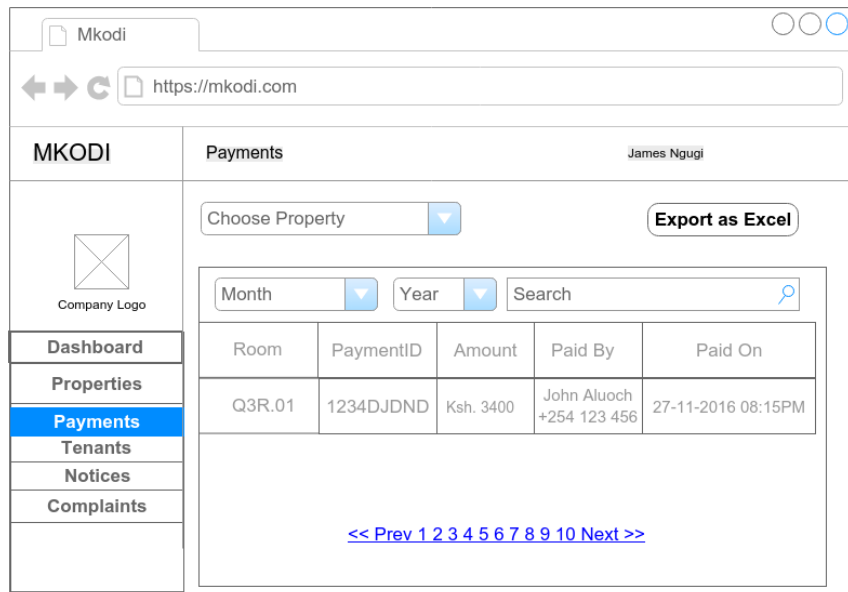
The user can then add rooms and tenants to the selected property as shown in this screen. Additionally, users can review all the complaints reported by tenants in their properties. These complaints are filterable by property or by searching for any keyword.

The user can also post notices, view any previously sent notices, their status, filter them by keyword and year. Finally, a user can see all payments made by tenants. Payment details are also filterable by date and searched by keyword.



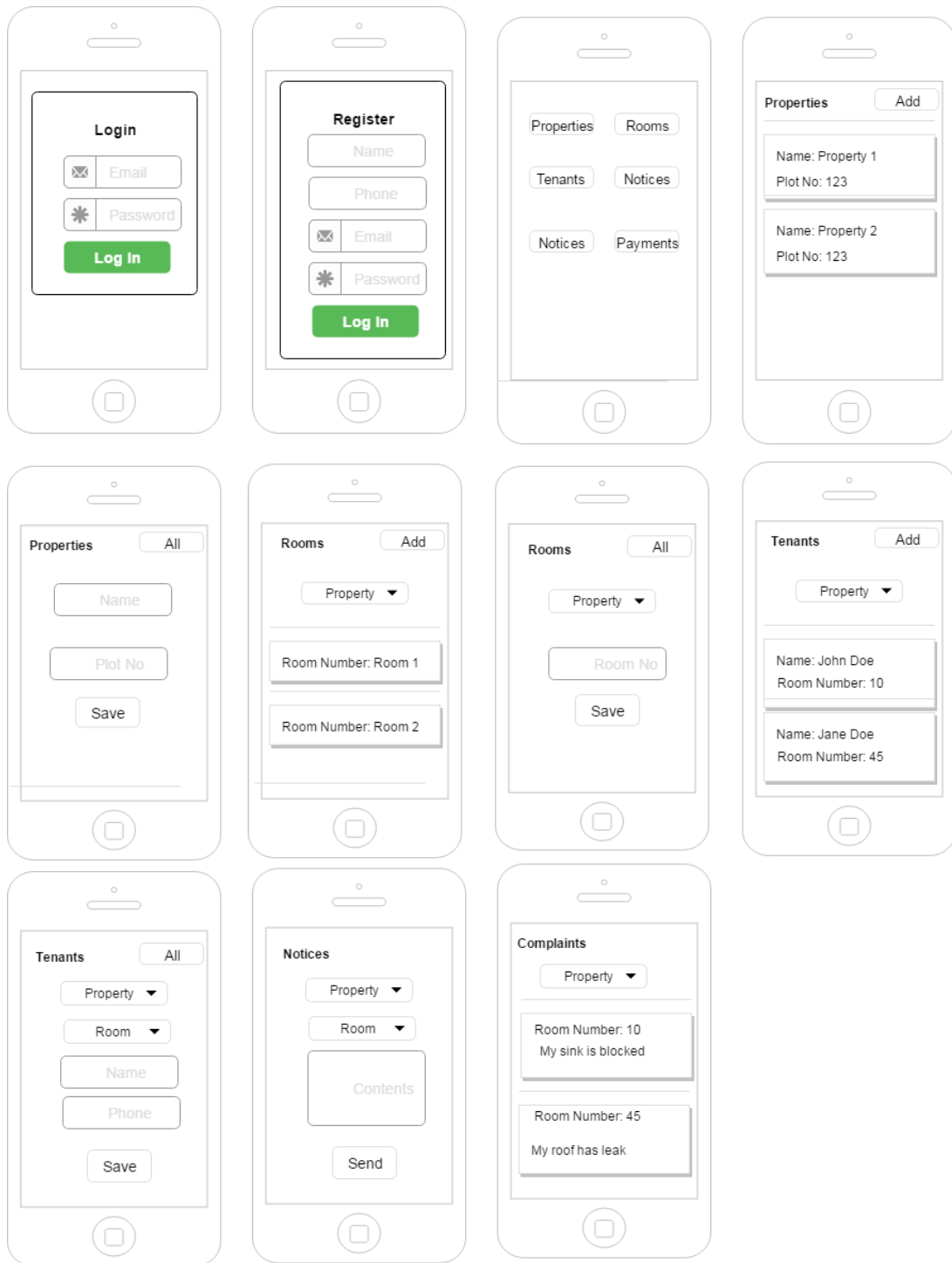
**Figure 4.10 Notices Page Wireframe**

The payments module shows the payments received as well as the paying tenants.



**Figure 4.11 Payments Page Wireframe**

## 4.9.2 Mobile Application



**Figure 4.12 Mobile Application Wireframes**

Figure 4.14 shows the mobile application wireframes.

### 4.9.3 Tenant USSD



**Figure 4.13 USSD Application Wireframe**

Figure 4.15 shows the tenant USSD application. The application offers tenants the ability to receive their account statement, and a make a payment. The application only allows payment of

amounts that fall within the threshold of MPESA transactions. After a user enters an amount, the tenant receives a USSD prompt initiated by MPESA, where they are required to confirm and finalize payment by entering their service PIN. They can also send complaints by SMS.

#### **4.10 Summary**

This chapter has discussed the general architecture of the proposed system as well as the algorithms used. It then discussed the system design tools used placing into context the business rules, network and security design. The application wireframes for the various interfaces were also illustrated and discussed. The next chapter discusses the system implementation testing procedures as well as the software and hardware environments used for system implementation.

## **Chapter 5: System Development and Testing**

### **5.1 Introduction**

The chapter focuses on the implementation and testing of the proposed system. The implementation explores different parts of the system, how they were implemented and the software as well as hardware environments under which they were implemented.

### **5.2 Software Environment**

Several technologies were used in the implementation of this project. MySQL 5.5 was the Database Management System (DBMS) used to implement the system's database. It was chosen because it was easy to use, highly customizable, yet low maintenance as a database management (Suehring, 2005). The back-end application was developed in PHP using the Laravel framework. The mobile application was developed using the Android framework. For SMS functionality, a third party SMS gateway was used. Messages are routed to respective tenant's phones with the help of SQL queries.

The mobile application was developed on the Android platform. The application communicates with the backend using the JavaScript Object Notation (JSON) data format. This format allows for easier exchange of data between the server and the application. This format is now supported by all Android devices and is far much simpler to use compared to other formats.

### **5.3 Hardware Environment**

The web application runs on a virtual server Ubuntu Linux server with two Gigabytes of main memory and twenty Gigabytes of storage. The mobile application runs on any Android device with at least 128MB of main memory and 100MB of storage. The USSD application is accessible by any GSM enabled device.

### **5.4 Modules of the System**

The application comprises of front-end and back-end subsystems: a front end USSD application, a front-end mobile application and a web application which is the back-end subsystem. PHP programming languages was used to develop the USSD Application. The backend web application used PHP and HTML 5. The mobile application was developed on the Android platform. These modules are discussed in details below:

### 5.5.1 USSD Application

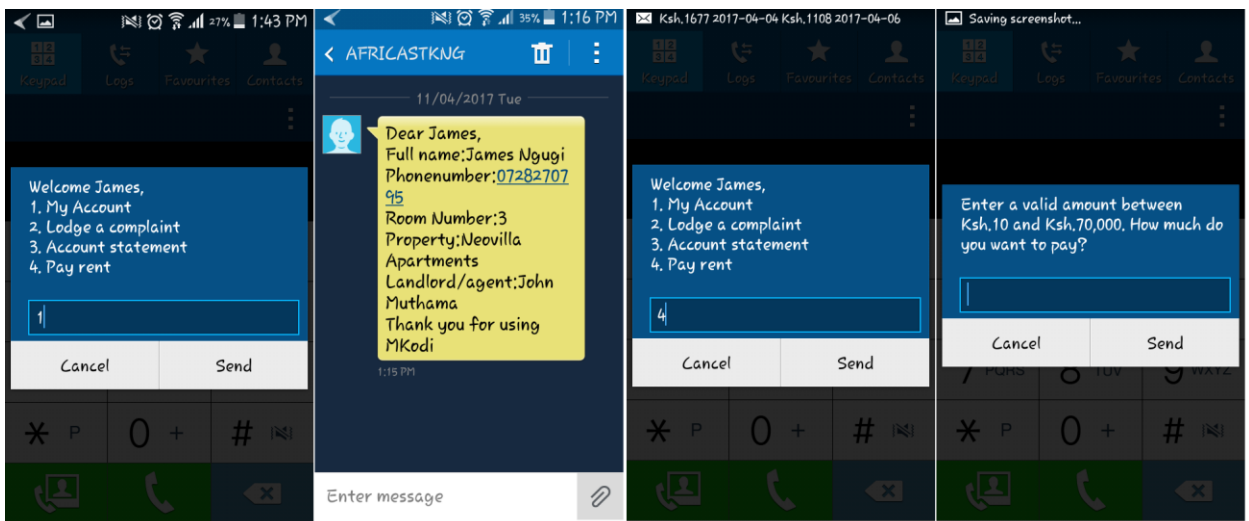


Figure 5.1 USSD Application Screenshot One

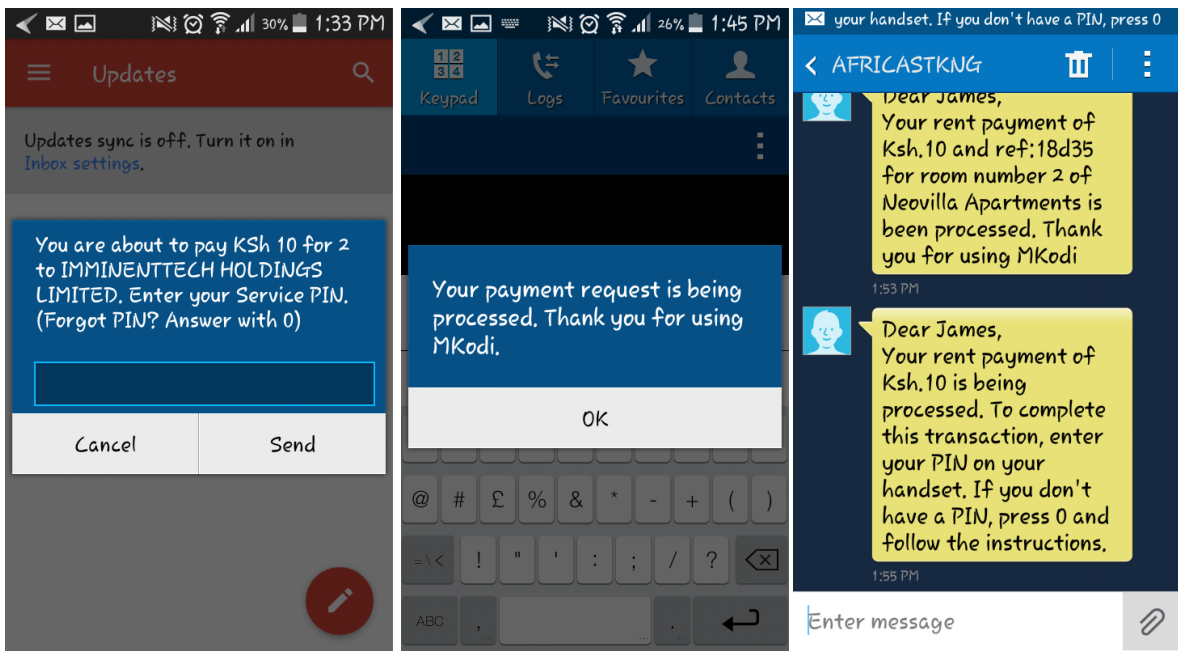
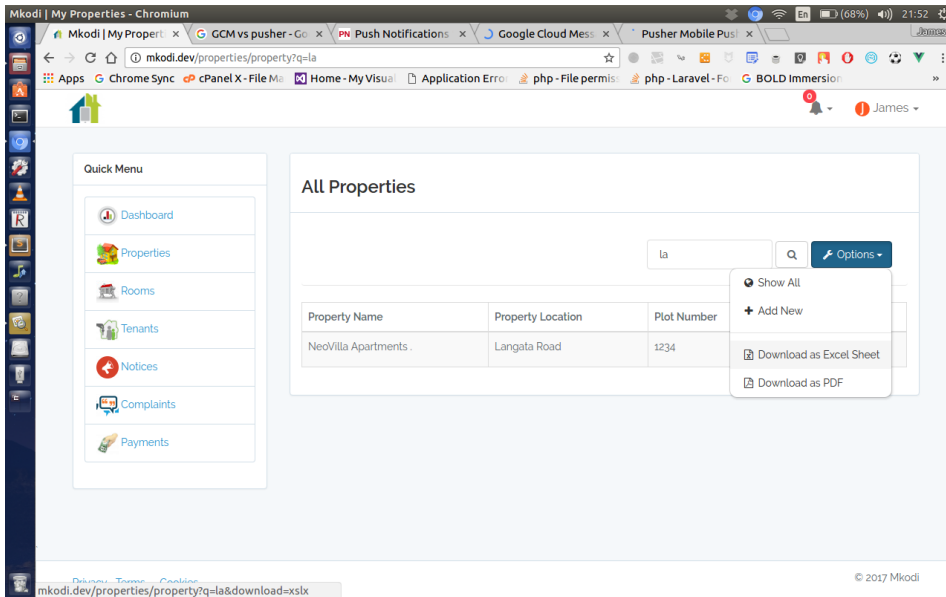


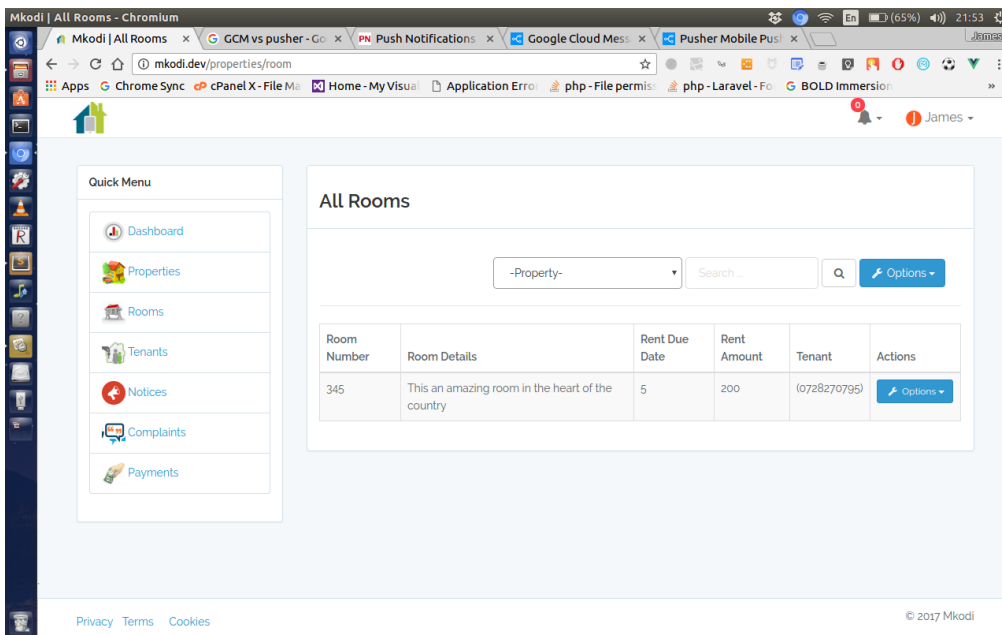
Figure 5.2 USSD Application Screenshot Two

### 5.5.2 Backend Web Application

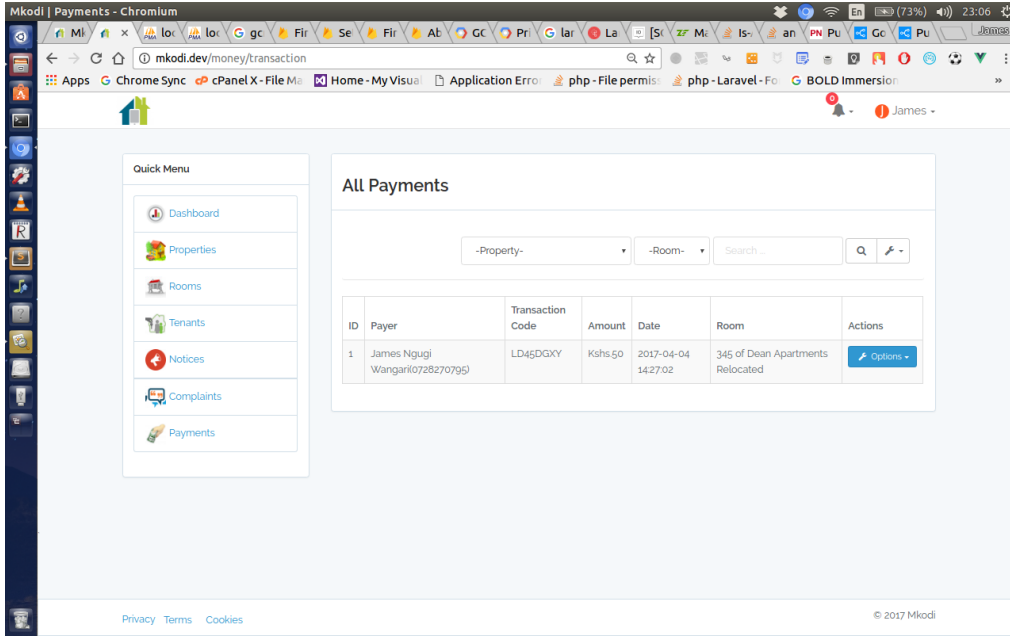
The backend application offers 4 main functionalities: property details management, tenant complaints reports and notice sending. The functionalities of these modules are shown in the screenshots below:



**Figure 5.3 Web Application Property Page Screenshot 1**



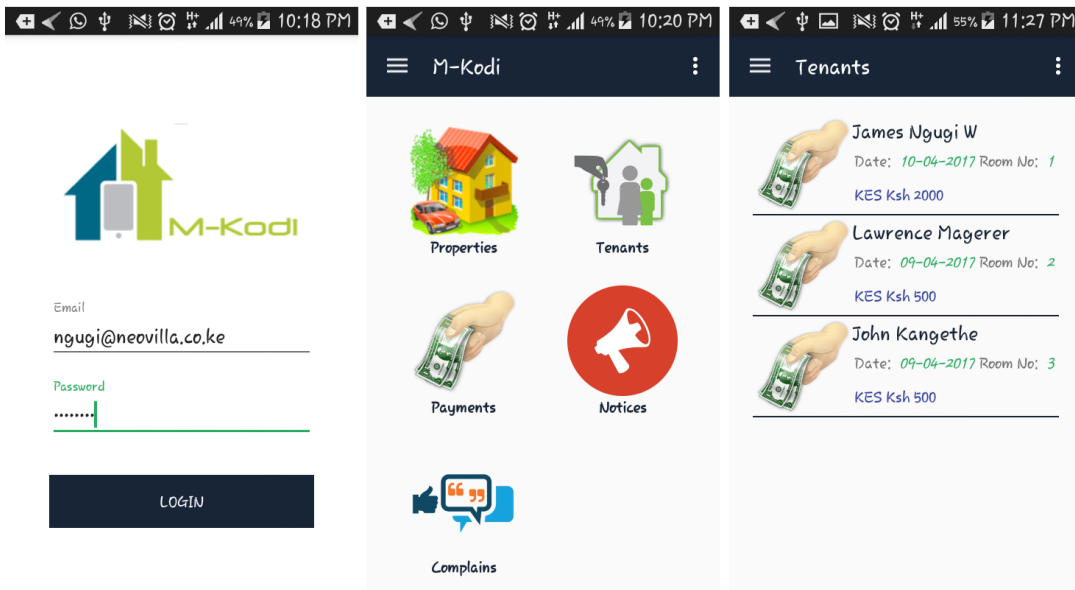
**Figure 5.4 Web Application Rooms Page Screenshot 2**



**Figure 5.5 Web Application Payments Page Screenshot**

As shown in Figure 5.3 and 5.4 the property module allows the creation, updating and deletion of property, room and tenant details. The payments module on the other hand lists and generates reports of all received payments. Other web application screenshots are attached in Appendix B.

### 5.5.3 Mobile Application



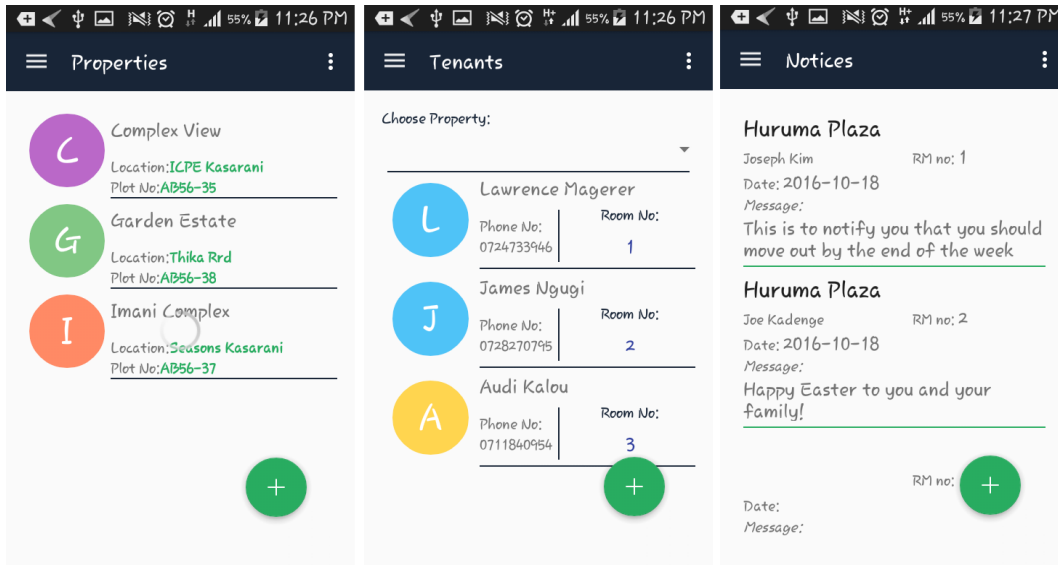


Figure 5.6 Mobile Application Screenshot One

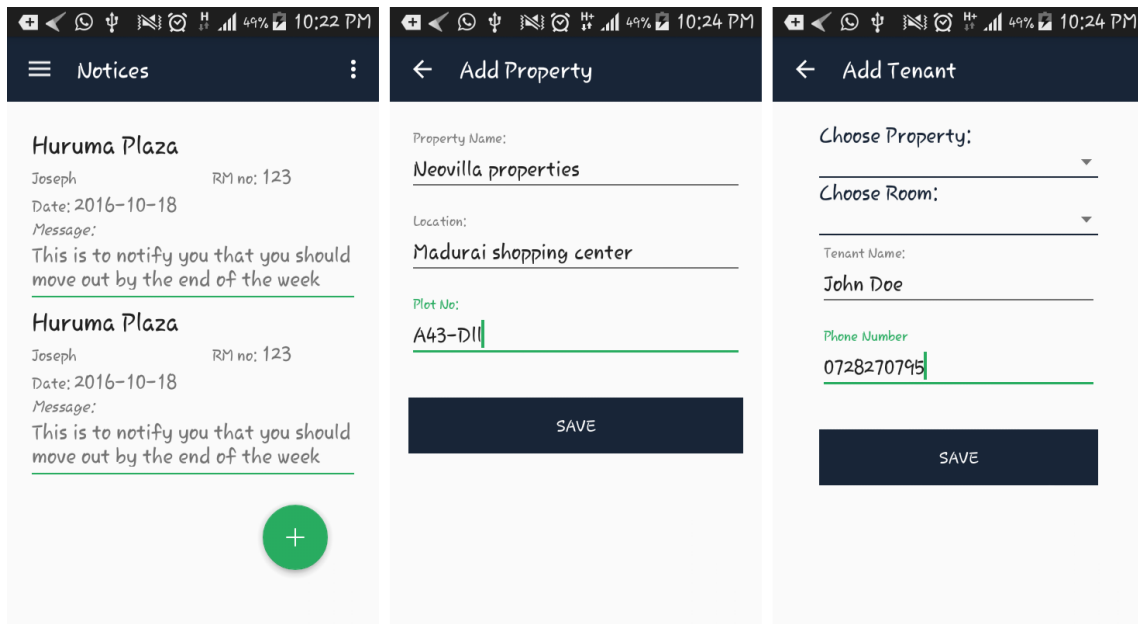


Figure 5.7 Mobile Application Screenshot Two

## 5.5 Testing And Validation

Usability and functionality tests were performed on the system. Functionality testing focused on the functional parts of the system which are: user login and logout, adding, editing and deleting

notices, properties rooms and tenants, viewing complaints, Tenants making payments and receiving notices. Usability testing evaluated the ease with which system users were able to achieve their system goals.

The application was given to property owners to validate ease of use and to test the final application. Testing was performed at several points in the system development life cycle as the application was being developed. All software features and combinations of software features to be tested were identified as well as those that were not supposed to be tested.

Unit testing was done where necessary by the researcher as opposed to the testers as it required detailed knowledge of the internal program design and code. In this scenario, writing unit tests using PHPUnit (Bergmann, 2017) tested individual software components or modules. All the modules passed this test. The following section provides a breakdown of the test results of the application by one property manager in areas of functionality, usability and compatibility.

### **5.5.1 Functional Testing**

This test was conducted to determine if all the modules and function areas of the application were working as expected. The areas tested as shown in Table 5.1 were the login and registration screen, property management module, payment processing module, the complaints module and notices module. From the indicated results in Table 5.1, all the system components were working as they were expected to be working and hence the application passed the functional test.

**Table 5.1 Functional Testing Results**

<b>Component</b>	<b>Description</b>	<b>Expected Behavior</b>	<b>Observed Behavior</b>	<b>Error</b>	<b>Verdict</b>
Payment processing module	Testing if the app allows payments by MPESA	The app should prompt the tenant to input the MPESA PIN and finalize the payment	The app should prompt the tenant to input the MPESA PIN and finalize the payment	None	Ok
Property management module	Testing if the app allows property managers to add property details	The app should accept and store property and room, tenant details.	The app should accept and store property and room, tenant details.	None	Ok
Complaints module	Testing whether the app accepts incoming SMS	The app should accept incoming SMS and create	The app accepts incoming SMS and creates complaints	None	Ok

	and create complaints from them.	complaints from them.	from them		
Notices module	Testing whether the app allows the creation and sending of notices	The app should allow the creation and sending of notices to tenants	The app should allow the creation and sending of notices to tenants	None	Ok

### 5.5.2 Usability Testing

This test was used to determine whether the flow of the application is user friendly from a user's perspective. Things like if a new user can understand the application easily, proper help documented whenever a user is stuck at any point were considered. Basically, system navigation was checked in this testing. From the indicated results in Table 5.2, the application was indeed user friendly and hence users could easily learn how to use the system. The application thus passed the usability test since it operates efficiently and effectively outside the application boundary with all interface systems.

**Table 5.2 Usability Testing Results**

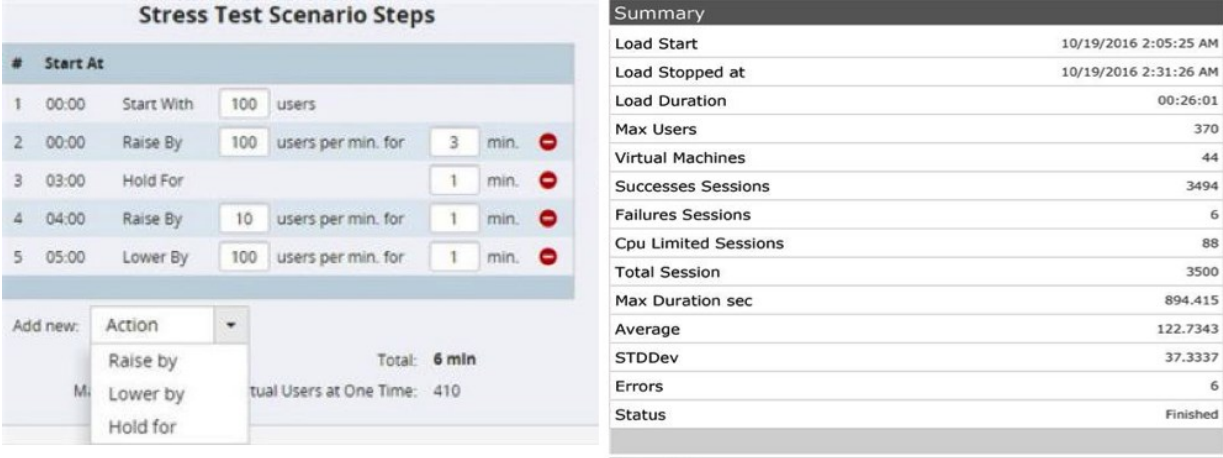
Test Area	Description	Expected Behavior	Observed Behavior	Error	Comment
Navigation	Testing if a user can move from one screen to another coherently without stress	A user should be able to move from one screen to another with much ease	A user is able to move from one screen to another with much ease	None	The application allows user to navigate to and from one screen easily
Error Messages	Testing if the application provides understandable messages in	The app should provide timely and understandable error messages	The app provides timely error and easy to understand error messages	None	The error messages are non-technical
Data Input	Testing for validation at while inputting data	The app should validate all inputs in the forms	The app validates all input as it is entered by the user	None	All inputs are validated
Recovery	Testing for	The app should	The app	None	Hardware

	recovery from failures and crashes	be able to restore itself to the last working mode	recovers well from crashes and other catastrophic problems		failures are minimal
--	------------------------------------	--	--	--	----------------------

**5.5.3 Load Testing**

This was done to check the amount of time the application takes to process a request. This test was conducted by sending a total of 3500 requests spread across a period of 40 minutes using a max of 370 users. The response rate was then be used to determine how many requests the application can handle during peak usage. The principal aim of this was to see how well the backend application scales well with increasing finance. As shown, there was only a small discrepancy between the actual expected numbers of users’ vs. the actual number of users sent to the application by the load testing software. Only 6 of the 3500 sessions failed, which is well within the 99% reliability threshold of most applications.

Figure 5.5 shows the default values used to setup LoadView and the results after running the test on the web application.



**Figure 5.8 Load Testing Results**

**5.5.4 Compatibility Testing**

Both the mobile and web applications were tested with different Android operating system (versions) and different web browsers respectively. This test was done to ensure the applications run on different Android and Web browsers for compatibility.

**Table 5.3 Mobile Application Compatibility Testing Report**

<b>Model (Android Version)</b>	<b>Category</b>	<b>Status</b>
Samsung Galaxy S3 (5)	High end device	Compatible
Samsung Galaxy mini (4.1)	Mid-range device	Compatible
Huawei U8185 (2.2)	Low end device	Compatible

**Table 5.4 Browser Compatibility Testing Report**

<b>Platform/Browser</b>	<b>Chrome</b>	<b>Firefox</b>	<b>Safari</b>
Android	Supported	Supported	N/A
iOS	Supported	Supported	Supported
Windows	Supported	Supported	Supported

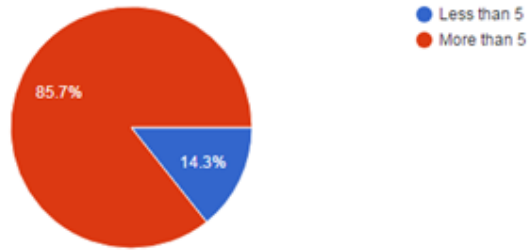
**5.5.5 Integration Testing**

The system was developed using a module-based approach. This test was therefore conducted on the system to know whether when the modules operated as desired when integrated together. This was done by combining all modules of the prototype and testing if they worked together was without any issue. Selenium tools were used to conduct regression testing to ensure the web application remained functional.

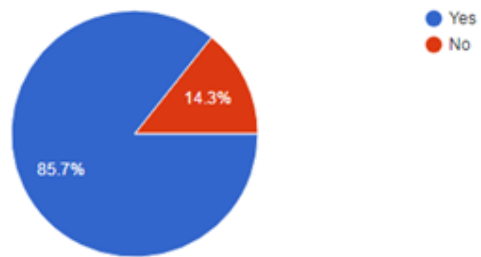
**5.5.6 Validation**

To validate the efficiency of the mobile application, the application was given out to 7 property managers who each managed more than 5 properties in the Madaraka area of the Langata constituency. They were allowed to use the application and later fill in a questionnaire to provide their feedback. This questionnaire can be found in appendix A.

How many properties do you manage? (7 responses)

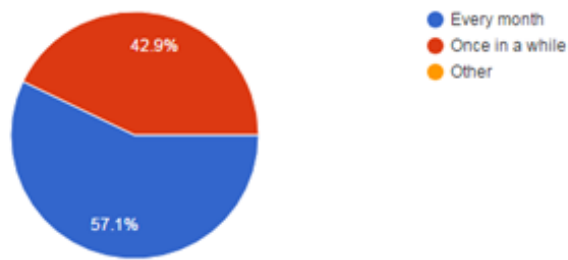


Do you find the application useful? (7 responses)



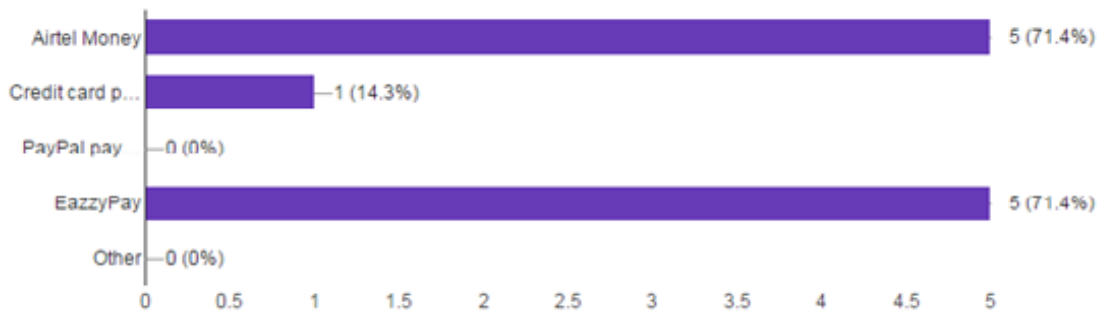
**Figure 5.9 Application Validation Questionnaire Results (a)**

How often will you use the application to collect rent? (7 responses)



Which other payment options would you want to be included on the application?

(7 responses)



What do you like about the application? (7 responses)

- It is simple to use, and it integrates well to the software we currently use: QuickBooks
- The interface is very simple. I get things done immediately!
- I like the website more..I have many properties to manage and they are easily navigable from the website.. the app is good but I like the website more
- The ability to send notices is a time saver.
- It's easy to use
- It is simple to use, but several things we need are lacking
- The payment actually works..That's all I need..

Figure 5.10 Application Validation Questionnaire Results (b)

### What do you dislike about the application? (7 responses)

Some features are missing. I want to be able to reply to user complaints from the Users application. This is currently not possible
Some features are missing. We use Quickbooks! There's no way to export from the app to Quickbooks without using excel.
Nothing specific
Some of our tenants are complaining that sometimes the user payment prompt times out before they can input their pin..we don't know how to fix that
It crashes when there is no internet access. Otherwise it works very well when I have internet
Mpesa allows only 70000 per transaction..some of our tenants pay more..They prefer credit cards rather than MPESA for such
No airtel money option

### Are there any other features you would like included on the application?

(7 responses)

please add the ability to respond to user complaints as well as allow tenants to attach images when reporting complaints
Allow Quickbooks exports!
Sending a notice to many recipients is a bit lengthy.. make it easier to send a notice to many recipients
Allow tenants to have airtel numbers or both safaricom and airtel
Add Equitel Money
Add more payment options to accommodate all rent ranges
Add the lipa na mpesa option..some of my tenants don't know what is an mpesa service pin

### Figure 5.11 Application Validation Questionnaire Results (c)

Figures 5.9, 5.10 and 5.11 show the results of the application validation questionnaire. Five of the seven respondents found the application useful. Additionally, some of the respondents proposed several changes to allow the interoperability of the application with the systems they currently use. These findings clearly show how effective the proposed application

## 5.6 Conclusion

This chapter has discussed the functions of the prototype, the software and hardware environments used to implement these functions. The testing procedure adopted in testing this application was also discussed in detail. In particular, all the modules of the proposed system were discussed in detail. The next chapter proceeds to discuss how the various objectives of this study were met and highlights any limitations that the proposed system encountered.

## **Chapter 6: Discussion of Results**

The system developed, as detailed in Chapter 5, was tested to ascertain whether it offered the required functionalities as identified in literature. This chapter, therefore, analyzes the findings made in this study, relating them with the research objectives and highlighting the merits and demerits of the proposed application.

### **6.1 Mobile-Based Rent Collection System**

This study developed a mobile based rent collection rent system. The mobile based rent collection system allowed tenants to pay for rent using mobile money at their convenience. As discussed in Chapter 2, mobile money services have gained wide popularity. By allowing tenants to pay for rent using MPESA, the system allows them to use a service they are well acquainted with. In addition, the system offers property managers the ability to track their tenants' payments in real-time. The transaction details can be downloaded as reports either as spreadsheets or as PDF. This allows property managers to incorporate these reports in the systems they are using currently with ease.

A critical part of this objective involved testing it to ensure it worked as required and offered the necessary functionality. The proposed application was subjected to functional testing, usability testing, load testing, compatibility testing and integration testing. The functions of the proposed application were functionally tested on a per module basis, where each module was tested to check whether it offered the intended functionality. The system was continuously subjected to unit tests as development progressed. These tests revealed flaws which were immediately rectified as development progressed.

As highlighted in Chapter 5, the application was offered to property managers for testing. Five of the seven respondents found the application useful. Additionally, some of the respondents proposed several changes to allow the interoperability of the application with the systems they currently use. These findings show how effective the property owners found the proposed application to be. In addition, the respondents proposed several new payment methods that they considered necessary for the application.

## **6.2 Advantages of the Application**

The first objective of this study was to identify the common rent collection methods. From reviewed literature, three main methods of collecting revenue/rent are currently use; payment by cash, bank deposit or through mobile money cash transfers. As discussed in section 2.4 of literature review, each of these has it's disadvantages. Cash payments were found to be the most common form of payment. However, cash payments are easy to lose and hard to trace. The property manager has to keep record of each tenant's payments. A review of literature went on to show how these methods do not provide any real-time information about each tenants payments to the property owners.

The second objective was to review existing mobile based rent collection systems. From the study findings, the researcher found that several information systems that allow property managers to collect rent from their tenants have been developed. These platforms support credit cards and PayPal as payment methods. However, few of these support mobile money as a payment method. Additionally, few of them allow access via USSD.

In the proposed application, since the property managers can send notices by SMS, these notices can be tracked to check which ones were delivered and which ones failed and why. Finally, tenants can access the service by USSD which works on both smartphones and low end phones. A great advantage of USSD is that it also works without internet access. This ensures that the service is accessible to all tenants who have access to a mobile phone.

## **6.3 Limitations of the Proposed Application**

Despite all the advantages highlighted in the previous section, some major limitations of the proposed application stand out. USSD application only allow a limited data size to be sent. Additionally, USSD sessions are short lived and don't allow long inputs. Thus although the MPESA payments are successfully initiated, the USSD push prompt sent by Safaricom times out before the tenant can complete the payment on their phone. Additionally, USSD applications cannot be accessible in areas out of network range or dead zones. This is greatly detrimental to tenants who travel a lot to areas outside their network coverage. Finally, the property manager's mobile application requires users to have access to a smartphone. This requirement locks out a great majority of property owners who don't have access to such devices.

## **Chapter 7: Conclusions, Recommendations & Future Work**

### **7.1 Conclusion**

This study reveals the challenges of the current rent payment methods. The findings made from reviews of current literature on the state of rent collection in Kenya have revealed gaps in the systems and payment methods supported. Key of these challenges includes lack of real-time information about tenant payments and limited support for mobile money as a payment method. Using the available technologies discussed in Chapter 4, led to the development of a mobile based rent collection system. This system was built adhering to the Agile Software Methodology. Agile Software Methodology enabled more frequent release with subsequent user feedback which led to development of a usable and reliable system. The system usability testing was performed thoroughly and respondents found it useful and satisfying. If this system is adopted it will allow tenants to pay rent using their mobile money at their ease and comfort. It will also allow property managers to monitor their tenants' payments, receive and respond to complaints by them faster and send notices to them as necessary. This will help reduce the challenges of rent collection.

### **7.2 Recommendations**

The mobile based rent collection system was of great importance to both tenants and property managers. The mobile application runs on a major portion of Android devices. The web application also runs on a major portion of devices available on the market. A key recommendation of this study is that the application be adopted property owners in order to allow the collection of rent.

In order to overcome the limitations identified in section 6.3, future studies can adopt the Lipa na Mpesa (FSD, 2012) system instead of the Mpesa C2B API used in this study. The Lipa na Mpesa system does not require a USSD prompt. Therefore, payments are unlikely to timeout, if at all. Additionally, in order to overcome USSD limitations, tenants can also be given access to a web application where they can make payments and access their account details. Such a platform would only require a working Internet connection.

### **7.3 Future Work**

However, the researcher noted that there was more that could be done to ensure rent collection is not as challenging. The application can be expanded to allow payment of rent through other mobile money services such as Airtel Money and EazzyPay. This will ensure that tenants have multiple ways of making payments.

The platform can also be advanced to allow tenants to receive alerts whenever their complaint has been resolved or is being worked on. This ensures that there is a continuously line of communication between the tenants and the property owners. The platform can also be extended to allow the sending of notices by other methods such email, Facebook messages or Twitter messages. These are platforms that most tenants are likely to be on. In addition, while this platform only offers functionality for property owners to access the web application, the platform can be extended to allow tenants to access the application, access and modify their profile details and view their payments. It would be interesting to see how the system would be used in other parts of the continent. Thus future replication of the study incorporating other areas would be eligible.

Finally, the application can be developed for the other platforms, Android, iPhone and windows so as to expand the market to all users. It is recommended that the app extends to all platforms to provide a more rewarding opportunity to all property owners. This can be achieved by use of cross-platform programming languages or doing integration to other platforms.

## References

- Alleman, J., & Rappoport, P. (2010). *Mobile money: implications for emerging markets. Communications and strategies.*
- Arvanitis, Y. (2015). African Housing Dynamics: Lessons from the Kenyan Market. *Chief Economist Complex | African Economic Brief*, 4(3), 5-7.
- Bergmann, S. (2017). *PHPUnit*. Retrieved January 10, 2017, from PHPUnit: <https://phpunit.de/>
- Bhanushali, P. (2016, August 8). *Design Patterns and Best code practices*. Retrieved April 21, 2017, from LinkedIn: <https://www.linkedin.com/pulse/design-patterns-best-code-practices-pratik-bhanushali>
- Bihis, C. (2015). *Mastering OAuth 2.0*. Packt.
- CBK. (2016). *2016 FinAccess Household Survey*. Nairobi: Central Bank of Kenya (CBK).
- Communications Authority of Kenya (CAK). (n.d.). *Kenya's mobile penetration hits 88 per cent*. Retrieved December 6, 2016, from Communications Authority of Kenya: <http://www.ca.go.ke/index.php/what-we-do/94-news/366-kenya-s-mobile-penetration-hits-88-per-cent>
- Cozy Services Ltd. (n.d.). *How Payments Work*. Retrieved February 9, 2017, from Cozy Service: <https://cozy.co/how-payments-work/>
- DeSmit, K. (2014, October 14). *The Best (and Worst) Ways to Accept Rent Payments*. Retrieved January 9, 2017, from LawDepot : <http://blog.lawdepot.com/the-best-and-worst-ways-to-accept-rent-payments/>
- Dolan, J. (2009). *Accelerating the Development of Mobile Money Ecosystems*. Harvard Kennedy School.
- Eberlin, E. (2016, October 17). *Options for Collecting Rent From Tenants*. Retrieved February 9, 2017, from thebalance: <https://www.thebalance.com/collecting-rent-from-tenants-2124980>
- FSD. (2012). Why doesn't every Kenyan business have a mobile money account? *FSD insights*(4).
- Gharegozi, A., Faraji, E., & Heydari, L. (2011). The Study of Information Technology Effect on E-commerce Growth. *International Conference on Advancements in Information* (pp. 50-54). Singapore: IACSIT Press.

- He, Y., Shi, Y., & He, A. (2010). *Research on Online Payment Mode of E-commerce*. Shandong Institute of Business and Technolgy, Yantai.
- Irungu, G. (2014, December 14). *Tenants to pay part of rent to KRA in new taxation plan*. Retrieved December 9, 2016, from Business Daily:  
<http://www.businessdailyafrica.com/Tenants-to-pay-part-of-rent-to-KRA/-/539552/2556276/-/fgnj9t/-/index.html>
- Janine, A. (2015). *Leapfrogging: a Survey of the Nature and Economic Implications of Mobile Money*. University of Oxford, Department of Economics, Manor Road Building. London: Oxford OX1 3UQ and Institute for New.
- Kemibaro, M. (2016, October 1). *Kenya's latest 2016 mobile & internet statistics*. Retrieved December 6, 2016, from Moses Kemibaro Pure Digital Passion:  
<http://www.moseskemibaro.com/2016/10/01/kenyas-latest-2016-mobile-internet-statistics/>
- Kimathi, V. N. (2011). *An exploratory study on the usage of m-payments by small and medium business enterprises in Kenya*. Nairobi: Strathmore University.
- Kinaku, K. (2011, October 26). *Kenya: The Convenience of Online Rent Payment and Collection Catching Up*. Retrieved December 9, 2016, from AllAfrica:  
<http://allafrica.com/stories/201110270028.html>
- KNBS. (2012). *Analytical Report on Urbanization, 2009 Kenya Population and Housing Census*. Kenya National Bureau of Statistics.
- KNBS. (2016). *Economic Survey 2016*. Nairobi: Kenya National Bureau of Statistics.
- Matinde, V. (2013, August 21). *Safaricom Launches House Rent Payments Service Similar to Manyatta Rent*. Retrieved February 26, 2017, from Techmoran:  
<http://techmoran.com/safaricom-launches-house-rent-payments-similar-to-manyatta-rent/>
- Mbiti, I., & Weil, D. (2011). *Mobile banking: The impact of M-PESA in Kenya*. National Bureau of Economic Research.
- Munassar, N. M., & Govardhan, A. (2010). A Comparison Between Five Models Of Software Engineering. *IJCSI International Journal of Computer Science Issues*, 7(5), 96.
- Mwakilishi. (2016, January 1). *What You Need to Know About Rental Tax in Kenya*. Retrieved December 9, 2016, from Mwakilishi:

- <http://www.mwakilishi.com/content/articles/2016/01/25/what-you-need-to-know-about-rental-tax-in-kenya.html>
- Nakhumwa, J. N. (2013). *Adoption of E-commerce Payment Systems by Commercial Banks in Kenya*. Nairobi: University of Nairobi .
- NCDDR. (2005). *What Are the Standards for Quality Research?* National Center for the Dissemination of Disability Research.
- Noppen, V. A. (2013). *The ABC's of Affordable Housing in Kenya*. Acumen Fund.
- Paycorp. (n.d.). *Introducing the easy way to pay rent and collect rent*. (Paycorp - The payment Solutions Company) Retrieved February 9, 2017, from Rentpay- Take control of your rent payments: <https://www.rentpay.com.au/agent/tour/>
- PayYourRent. (n.d.). *Collect Rent Online*. Retrieved February 9, 2017, from PayYourRent: <https://payyourrent.com/managers/>
- Provos, N., & Mazières, D. (1999). A Future-Adaptable Password Scheme. *1999 USENIX Annual Technical Conference*. Monterey, California: The USENIX Association.
- RentPayment. (n.d.). *RentPayment*. Retrieved February 9, 2017, from Collect & Pay Rent online with RentPayment: <http://www.rentpayment.com/>
- Suehring, S. (2005). *MySQL Bible*. Wiley Publishing Inc.
- UN-HABITAT. (2003). *Rental Housing An essential option for the urban poor in developing countries*. Nairobi: United Nations.
- William, J., & Suri, T. (2011). Mobile Money: The Economics of M-PESA. *BNER Programs*.
- Xinhua. (2014, October 14). *Landlords in Kenya now shunning property agents*. Retrieved December 9, 2016, from Standard Digital Media: <https://www.standardmedia.co.ke/article/2000138202/landlords-in-kenya-now-shunning-property-agents>
- Yifei, Z. (2008). *Research on Online Payment Pattern and Security Strategy of E-commerce*. Henan University of Urban Construction, Pingdingshan, Henan, China.
- Zend Corp. (2017). *PHP Data Objects*. Retrieved January 17, 2017, from PHP: <http://php.net/manual/en/book.pdo.php>

## Appendix A: Application Evaluation Questionnaire

Please fill out the following questionnaire to help us understand more about how you use the mobile based collection system. Thank you for your participation.

Name: ..... Profession: ..... Date: .....

### *Section A – Main Questions about the Application*

1. How many properties do you manage?
  - a) Less than 5
  - b) More than 5
2. Do you find the application useful?
  - c) Yes
  - d) No
3. How often will you use the application for the collection of rent? YES
  - a) Every Month
  - b) Once in a while
4. Which other payment options would you want to be included on the application?
  - a) Airtel Money
  - b) Credit card payments
  - c) PayPal payments
  - d) EazzyPay
  - e) Other:

---

5. What do you like about the application?
-

---

---

5. What do you dislike about the application?

---

---

---

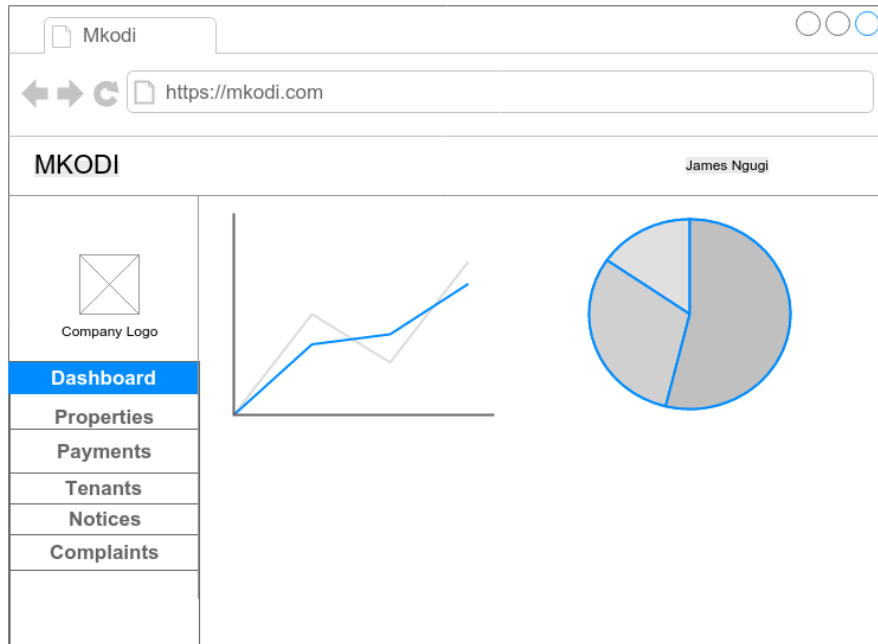
6. Are there any other features you would like included on the application?

---

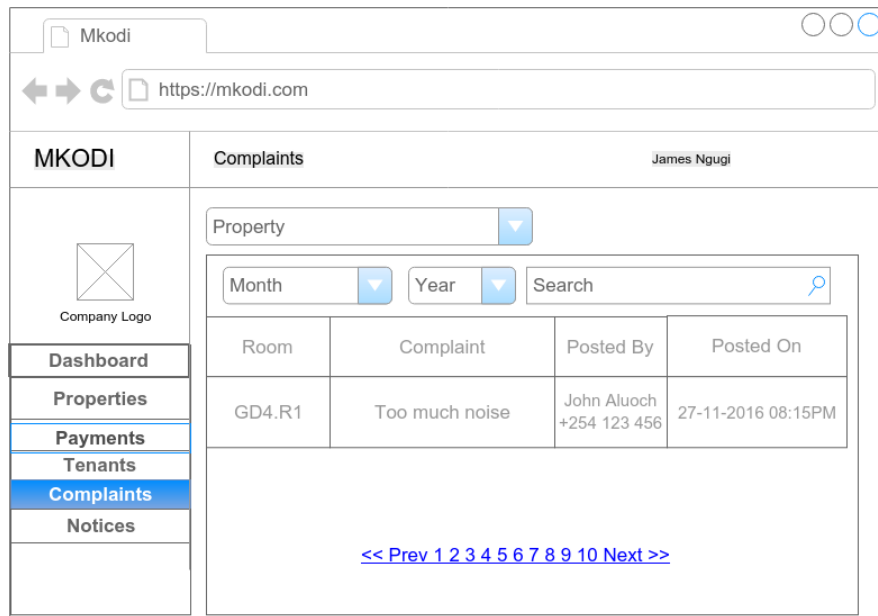
---

---

## Appendix B: Extra Application Wireframes

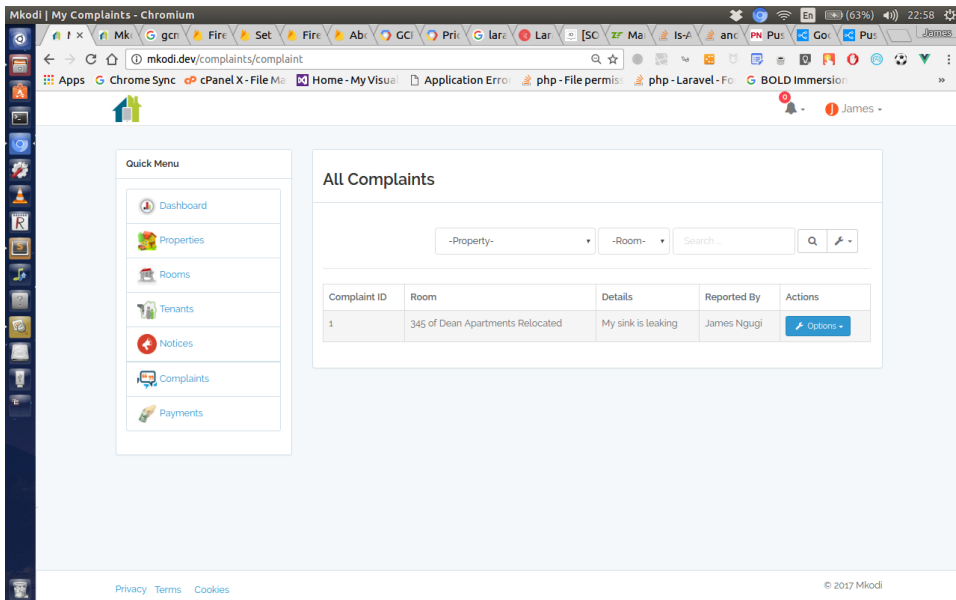


### Appendix B.1 Web Application Backend Wireframe

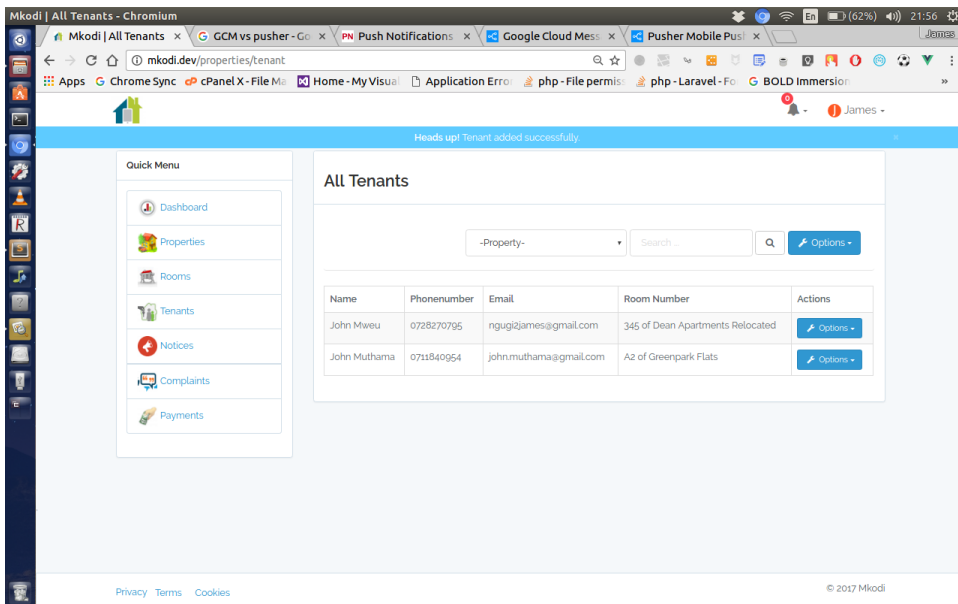


### Appendix B.2 Complaints Page Wireframe

## Appendix C: Extra Application Screenshots



### Appendix C.1 Complaints Page Screenshot



### Appendix C.2 Tenants Page Screenshot

# Appendix D: Turnitin Report

The screenshot displays the Turnitin interface for a document titled "A Mobile Based Rent Collection System" by James Ngugi Wangari. The document is identified as "new stuff" by KATE MOROKA. The overall similarity score is 16% (SIMILAR). The document content includes the title, author name, ID (092499), and a research proposal title: "A Research Proposal Submitted in Partial Fulfilment of the Requirements of the Degree of Masters of Science in Mobile Telecommunication and Innovation (MSc. MTI)". A red highlight and a small red square with the number '1' are visible above the first line of the research proposal title. The 'Match Overview' sidebar on the right lists the following matches:

Rank	Source	Similarity
1	Submitted to Strathmor... Student paper	3%
2	Yongqiang He. "Resea... Publication	2%
3	hubasia.org Internet source	1%
4	stormpath.com Internet source	1%
5	www.cit.mak.ac.ug Internet source	1%
6	www.easylaravelbook.c... Internet source	1%
7	Submitted to Saint Pau... Student paper	<1%
8	www.ejobm.org Internet source	<1%

The interface also shows navigation options (Originality, GradeMark, PeerMark), a search bar, and a page indicator (PAGE: 1 OF 89). The bottom right corner indicates "Text-Only Report".

## Appendix D.1 Turnitin Report